An emotion-inspired anomaly detection approach for cyber-physical systems resilience

Eskandar Kouicem, Clément Raievsky, Michel Occello

An emotion-inspired anomaly detection approach for cyber-physical systems resilience

Eskandar Kouicem[1], Clément Raïevsky[1], and Michel Occello[1]

Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
{eskandar.kouicem,clement.raievsky,michel.occello}@univ-grenoble-alpes.fr

Abstract. Nowadays, cyber-physical systems (CPS) are becoming ubiquitous in various application domains. The variety of design and implementation methodologies utilized for cyber-physical systems, as well as the dynamic interaction of its components, make the resilience of these systems a major challenge. We aim to increase the resilience of these systems in a decentralized way by leveraging knowledge of the social sciences and humanities (SSH) and especially emotional processes. Both individual decision-making processes and social coordinating mechanisms are based on emotional inspiration. Our hypotheses and studies on resilience approaches, cyber-physical systems and emotional processes allowed us to choose the multi-agent paradigm. In this paper, we present the results of our research on resilience, which includes an emotion-inspired anomaly detection approach for improving CPS resilience. This approach is integrated into an agent architecture, compared to the literature, and validated through the development of proof-of-concept scenarios. The experimental results prove its advantages in terms of resilience properties.

Keywords: Anomaly detection · Resilience · Cyber-physical systems · Agent architecture · Multi-agent systems · Artificial emotions.

1 Introduction

Almost all complex systems are controlled by computers that interact with the real world. These interactions are not done through a touch screen, mouse or keyboard, but also through direct actions in the physical world. These systems are made of interconnected subsystems, at least some of which interact directly with the physical world, which is why we call them "*cyber-physical systems*".

The development of interoperability protocols, lower hardware costs, and the simplicity with which a variety of hardware components can be connected provide designers with a wide number of configurations and combinations of components over which they may not have complete control. Furthermore, most of these systems are designed to be "open". As a result, the system's designers will be unable to anticipate all possible scenarios. This concerns the systems' resilience, or its ability to detect, manage, and adapt to specific or unusual situations that the designers may not have anticipated [22].

In previous works [12,13], we draw on knowledge from the human and social sciences, particularly emotional processes, to design an agent architecture that

improves the resilience of cyber-physical systems. In this architecture, we have integrated processes for detecting abnormal situations. The main objective of this paper is to present our anomaly detection approach used in this architecture.

In this paper, section 2 defines resilience and classify relevant work on anomaly detection for CPSs resilience, as well as a position within that work; section 3 presents the emotion-inspired anomaly detection approach and the R-ECM architecture (R: Resilience, E: Emotional processes, C: CPS and M: MAS); Section 4 demonstrates the viability of the proposed approach; Section 5 presents our conclusions and relevant future directions.

2 Related work

In this section, we define resilience and provide a classification of anomaly detection approaches that address it.

2.1 Resilience

Resilience is studied by researchers from various fields. In psychology, resilience is defined as a person's or a group's ability to grow and project themselves into the future overcoming destabilizing events, tough living conditions, and often severe trauma [19]. Resilience in systems engineering refers to how quickly a system bounces following an incident that causes its degradation. It is defined by the computer networking community as a combination of reliability and tolerance. Resilience is defined by the IT community as the continuity of service delivery and the availability of functionality [20]. For our work, we have adopted the definition of Woods [22], from the resilience engineering field: "Resilience is the ability to recognise and adapt to handle unanticipated perturbations that call into question the model of competence, and demand a shift of processes, strategies and coordination."

Resilience in artificial systems should be distinguished from robustness, which is closely related to it. Because of its design, a robust system can "resist" to abnormal situations by maintaining its performance without affecting its functionality. A resilient system, on the other hand, can *detect* abnormal situations and adapt its operations to keep its critical functions [14]. So robustness does not allow the system to adapt its behavior, but resilience does.

2.2 Anomaly detection for CPS resilience

Anomalies and faulty components must be detected in order to preserve the system's resilience and provide correct operation. A study of existing detection approaches for resilience yielded the following four categories, in the table 1.

In [2], authors use redundancy to merge data from different sensors and simultaneously calculate trust values for the information sources in VANETs.

Falcone *et al.* [9] explains how to improve resilience using in-the-field runtime approaches. Autonomic monitors use sensor data to observe, analyze, and plan

Table 1. Classification of detection approaches for CPS resilience.

Category	Key words	Application areas	Features	Examples
Redundancy based	Additional information sources, Triple Modular Redundancy (TMR).	Avionics, automotive ECUs, VANETs.	Reliability and multiplying components.	[2]
Monitoring based/ dependability based	Machine learning, Signal Temporal Logic	CPS.	The normal behavior of the system needs to be planned in advance.	[11,9]
Statistical model based	Gaussian models, histograms, machine learning, data mining, deep learning.	CPS, Iot, time series...	Simplicity, suitable for time series and used in CPS.	[15,6]
Signature based	Intrusion detection, functional footprint.	Malware detection, IoT.	Efficient (few false positives), requires accurate signature parameters.	[8,10]

before taking action to detect anomalies. They respond to failures by using redundancy or variants. In [11], authors build a signal temporal logic (STL) formula using the data that represents the system's usual behavior. Abnormal traces are those that do not conform to the formula.

Statistical models are used to detect anomalies in [6,15]. These approaches, which use K-means, clustering, machine learning, and deep learning to find anomalies in time series, are effective. The nature of statistical model used is determined by the time series' complexity.

A signature-based approach is used by [10]. This approach efficiently detects intrusions with a low amount of false positives, but it requires a well parameterized signature. Machine learning can help to reinforce it [8].

According to our classification, many centralised or redundancy-based approaches require reliable communication and may face the "Single point of failure" problem. The designer of monitoring-based approaches must anticipate all the situations that the system will face during its normal operation. Since most CPSs are designed to be "open", he will not be able to anticipate all possible scenarios when a component is added or removed.

To avoid these issues, we use incremental processes which can rely on component collaboration to detect anomalies. A component initiates anomaly detection, following which the same component or other components, depending on the situation, trigger other processes. In our approach, perceived data is represented as time series. Due to the benefits of simplicity and speed, we have chosen to use a basic statistical model in our detection approach to process this type of data. To improve its efficiency, we combined it with a signature-based detection approach.

3 The proposed approach

As mentioned previously, anomaly detection is a series of incremental processes in our approach.

To begin, we apply a basic statistical model to find anomalies in the sensor's data (out-of-domain values, strange sequences, or long repetitions of similar data). This choice depends on the simplicity and quickness with which this statistical model provides results, as well as the fact that it is not resource expensive. At first, each sensor elicits this process on its own, then correlates it with other sensors. A *perception grid* is used in this process. "A situation has no significance within itself; each individual has their own perception grid that determines whether the situation is good or bad [1]". In other words, sensor's perceptions are interpreted differently depending on its environment model. If the agent has interpreted a perception as anomaly, the result of this process is "a doubt". It refers to the occurrence of a situation which triggers the emotional episode in emotional processes. In case of doubt, the sensor must continue its perception functions, so it will increase its sampling frequency and remain more vigilant by ignoring any doubtful perception. This is known as "the arousal" in emotional processes.

Then, if there's any doubt, it move on to the next process. The sensor's interpretations will be based on the perceived data history and its *episodic memory*, rather than the environment model. In cognitive psychology, episodic memory is the mechanism through which a person recalls past experiences along with their context (date, location, and emotional state) [21]. This memory is used to store the normal and abnormal situations that the sensor encounters. This allows the sensor to learn its operating signatures. In emotional processes, this process refers to "the appraisal" of the situation. In psychology, appraisal is the process of extracting emotions from evaluations of events that produce specific reactions in different persons [18].

Finally, if the appraisal does not validate the doubt, the detection will be assisted by the system's similar sensors. If the situation is confirmed as abnormal by their appraisal, the other sensors lower their tolerance thresholds and communicate the result of the appraisal as a confirmation or rejection to the sensor which first detected the anomaly. This sensor's episodic memory is improved by creating an episode that reflects the negotiation result, and then its tolerance thresholds and perception grid are updated. The sensor will better preserve the system's functions with this upgrade by identifying and isolating the disturbance faster in the case of a similar situation in the future. In this process, some of the system's operating parameters have been changed, resulting in a change in readiness to act, which refers to the activation of a behavioural script in the emotional processes (but not its realization) with the goal of changing the individual's relationship with his environment and focusing his attention on more important things.

In order to integrate these processes to the components of a CPS, we chose the multi-agent paradigm. This choice is justified by the fact that MAS is built to accommodate for the distributed nature of CPSs. It's especially well-suited

to various resilience-related issues, such as the "single point of failure". The agent's autonomy distinguishes our approach from those based on redundancy or centralization, which are frequently used in resilience approaches [17,7]. Agent-centered design is also interesting for our approach because of its ability to integrate knowledge from SSH such as emotions.

After a study of agent architectures, we chose to integrate our decision-making processes in a layered agent architecture to organize them [16]. This architecture integrates reactive architectures' simplicity, low algorithmic complexity, and fault tolerance [5] with more cognitive architectures' capacity to exploit non-local information, learning capabilities, and social interactions [3]. However, managing the interactions between the different layers to achieve the intended behavior is a challenge in this type of architecture.

The decomposition of cognitive processes into layers allows us to implement simple, perception-related, emotion eliciting mechanisms in a reactive layer while allowing higher level cognitive processes, potentially based on symbolic information and reasoning, to unfold without interrupting processes supporting critical functions [4].

Figure 1 illustrates the R-ECM architecture, which integrates emotional processes (in blue) that allow CPSs to become more resilient [13].

This architecture is divided into two parts: on the left, which is made of the agent's processes and behaviors, on the right, which is made of the agent's knowledge. It is also important to note that the layers are finite-state machines with the following functions: The reactive layer (RL): ensures reliable behavior on short time scales by interacting with the environment, perceiving data, and utilizing a perception grid (PG) to recognize major events and potentially abnormal situations before transmitting the data to the proactive layer; The proactive-deliberative layer (PDL): works on a longer time scale and is used to initiate specific behaviors like evaluating the reactive layer or the other agents' doubt (*the appraisal*). It takes decisions according to its data, its knowledge, its action plans and the episodic memory (M); The social layer (SL): uses social relations knowledge (SR) to communicate with other system agents, as well as the diffusion and negotiation of detected situations.

The *message exchange protocol* is used to communicate between the layers and the agents. The actions in the R-ECM architecture are not always triggered by a perception. The proactive-deliberative layer can also initiate actions in response to internal decisions, while the social layer can initiate actions in response to a message from another agent. The functions and interactions of the layers are illustrated in figure 3. In our previous work [13], we well explained the functioning of the architecture and its layers.

4 Experiments

One of the main goals of this paper is to show how our detection approach and the R-ECM architecture work, as well as the added value provided by decisions made about the layered agent architecture on the one hand, and processes, knowledge,

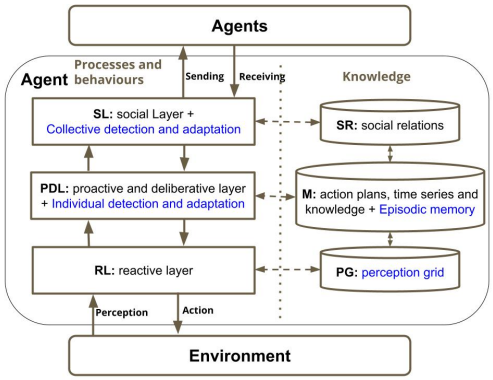

Fig. 1. R-ECM architecture [13].

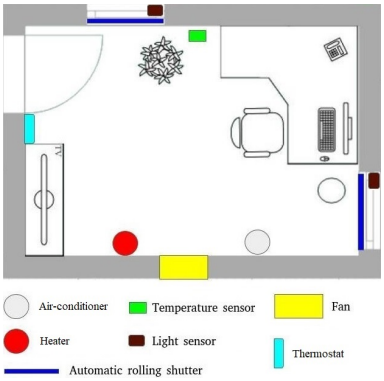

Fig. 2. Components in a room.

and behaviors on the other. We also want to validate our choice to combine a simple statistical model, a perception grid, an episodic memory, and a collective detection for detecting anomalies.

4.1 System description

To demonstrate the functionality of the processes and the agent architecture, we used a building temperature control system "CPS_{btc}" (see figure 2). In CPS_{btc}, the components (sensors and actuators) are implemented as R-ECM agents, which can communicate via messages, have some autonomy, and store their own data. We use T_i as a name of room i temperature sensor, Ac_i and He_i for air-conditioner and heater. In our multi-agent system, we'll have several groups of agents, each one made of agents from the same room. In our system, another sort of agent organization is taken into account: the organization of similar neighboring agents. Layers, agents and the environment are implemented as Java Threads that work permanently.

4.2 Measures of resilience

We used several quantitative measurements to evaluate our approach. Our assessment measures are mainly based on the resilience features mentioned in [13] which are: critical functions preservation, reactivity in terms of anomaly detection, anomaly sensitivity and the impact of the approach on system resources.

According to these features, we chose to log following measures in the simulations: perceived temperature (for CPS_{btc}), number of exchanged messages, CPU usage, memory usage (RAM), and anomaly detection delay.

4.3 Scenario description

Before describing the scenarios, we first define some notions that concern our simulations:

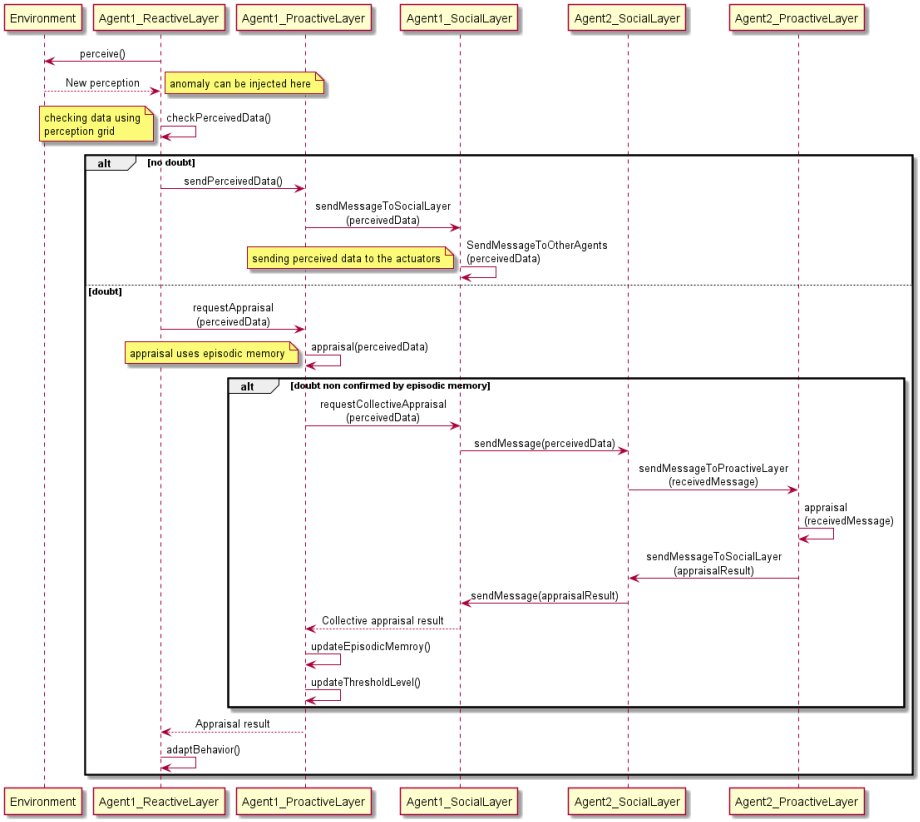

Fig. 3. The functioning of the anomaly detection in R-ECM.

- A faulty room: a room with faulty or abnormal behavior.
- Anomaly injection: an action made by a faulty room, it can send an abnormal value in response to a reactive layer perception.
- Abnormal value: a temperature value that has a significant deviation from the current temperatures.
- Anomaly injection probability: a value between 0 and 1. This value is used by the faulty room to decide whether it injects an anomaly or not.
- Time step: in our simulations, one step corresponds to a 30-minute system, so one-day simulation corresponds to 48 steps. In each step, the environment is updated and several decision cycles and message exchanges take place.

Scenario 1: To demonstrate that our detection approach allows the R-ECM architecture to maintain crucial functionality without impacting system resources, we used the following scenario: the faulty room injects the agent T_i with abnormal temperatures with a probability of 0.01; the agents Ac_i and He_i must receive the right temperatures in order to ensure the desired temperature (of the

agent Th_i); after 24 steps (at 12H system time), the faulty room increases the anomaly injection probability to 0.02.

Scenario 2: To illustrate that our detection approach provides agents with autonomy and anticipation, as well as how a layered agent architecture allows for action parallelization at the system and agent levels. We compare our approach with a combination of redundancy and monitoring [9]. This approach replaces the emotional processes in the R-ECM architecture. In the proactive-deliberative layer of actuators, we implement an autonomic monitor for detecting anomalies; it reacts to failures by switching the temperature sensor, using redundancy. A redundant temperature sensor ($T_{i.1}$ and $T_{i.2}$) is deployed in the faulty rooms to illustrate this scenario: after 10 steps (at 5H system time), the faulty room plants the $T_{i.1}$ agent so it no longer makes perceptions; the agent $T_{i.2}$ will provide the temperatures to the room's actuators; the actuators of room i will therefore base their actions on the perceptions of $T_{i.2}$ only; at 12H system time, agent $T_{i.1}$ will be restarted and anomalies will be injected to agent $T_{i.2}$ between 14H and 18H.

4.4 Results and evaluation

In simulations with 3 rooms (27 agents for scenario 1 and 28 agents for scenario 2) we obtained the following results: For scenario 1 with our approach, we got three injected anomalies, the first one injected at 5:00, detected by the reactive layer at the same step and confirmed by other agents two steps later. The time needed to get the confirmation from all agents is 850 ms (the anomaly detection delay). The second one injected at 08:00, detected by reactive layer and confirmed by the episodic memory in the same step. The time needed to get the confirmation from episodic memory is 156 ms. The third injection at 22:00, detected by the reactive layer at the same step and confirmed by other agents two steps later. The time needed to get the confirmation from from all agents is 950 ms.

For scenario 2 with our approach, an anomaly was injected at 15:00, detected by the reactive layer one step later and confirmed by other agents two steps later. The time needed to get the confirmation from all agents is 859 ms. With the autonomic monitoring approach, two anomalies were injected. The first one injected at 14:00, detected by the monitor one step later. The time needed to detect this anomaly is 549 ms. The second one injected at 16:00, detected by the monitor one step later. The time needed to detect this anomaly is 398 ms. We conclude that the collective detection is slower than appraisal and autonomic monitoring. But the appraisal using episodic memory is still faster than autonomic monitoring in term of the anomaly detection delay.

In figures 4 and 5, we see that injecting anomalies did not impact the operation of the system using our approach and autonomic monitoring. The agents using both approaches were able to maintain the desired temperature (20 C°) when they detect the anomalies.

We can see in scenario 1 (figure 6) that the first anomaly between 04:00 and 05:00 has increased the amount of exchanged messages, which is justified by

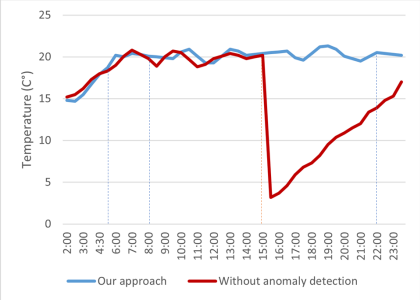

Fig. 4. Perceived temperature in faulty room using scenario 1.

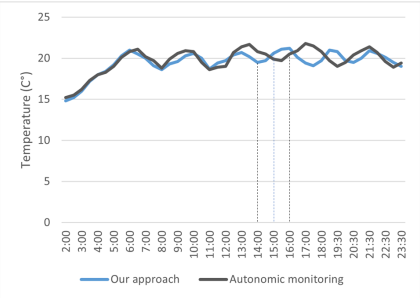

Fig. 5. Perceived temperature in faulty room using scenario 2.

the fact that the appraisal requires more message exchange than normal system operation. Because the reply was so quick due to episodic memory, the second anomaly at 08:00 did not increase the amount of messages exchanged. The third anomaly at 22:00 has increased the number of exchanged messages.

When anomalies are injected in both approaches, there are some gaps in scenario 2 (figure 7), but the gaps in our approach are more significant because the layers and agents exchange messages once they have a doubt about a perception. We conclude that our approach has very little effect on the amount of exchanged messages.

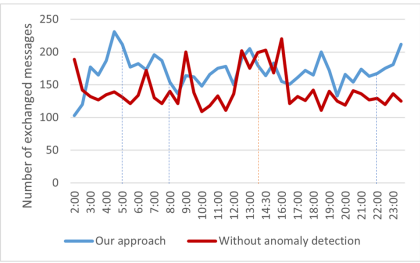

Fig. 6. Number of exchanged messages step by step using scenario 1.

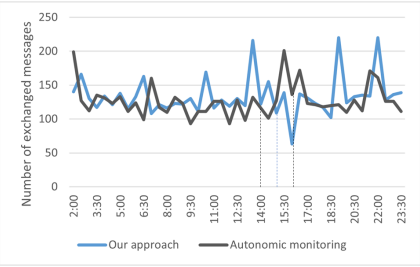

Fig. 7. Number of exchanged messages step by step using scenario 2.

In figure 8, when the simulation was launched (before 10:00), the execution time of the steps was longer than in the end. So the anomalies occurrences impact the running time but not a lot compared to the simulations without processes. In figure 9, there the running time is close for both approaches. Because of the incremental processes, there is a big difference (30 cycles) between our approach and simulations without anomaly detection (see figure 10). Same comment for figure 11, we just add that there is a big gap when an agent is faulty. In our

approach, the incremental processes for detecting anomalies are integrated in sensors but this is not the case for autonomic monitoring.

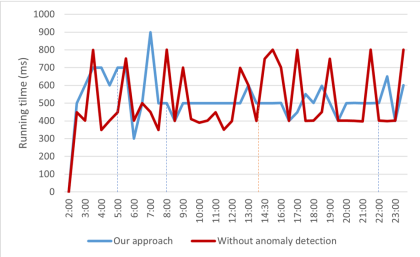

Fig. 8. Simulation running time step by step using scenario 1.

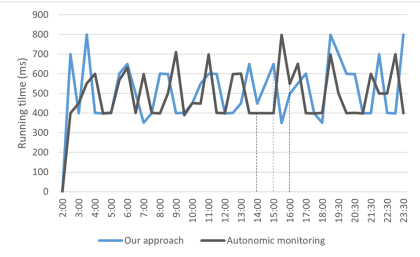

Fig. 9. Simulation running time step by step using scenario 2.

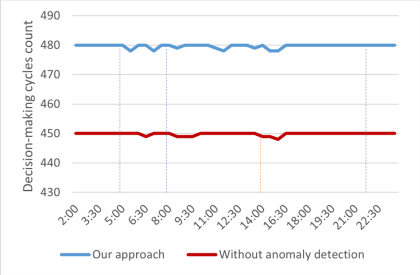

Fig. 10. Decision-making cycles count step by step using scenario 1.

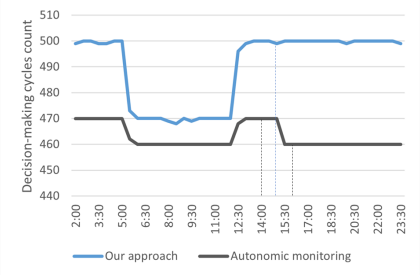

Fig. 11. Decision-making cycles count step by step using scenario 2.

In figure 12 and 13, we measured the time spent by the processor in the simulations. As we can see, our approach had no significant impact on CPU time. We note that the autonomic monitoring approach is more CPU-intensive. In figures 14 and 15, we measured the used memory (RAM) in each step of the simulation. We have drawn the same conclusion as the CPU time. But we note that the the autonomic monitoring approach saves a little bit of memory.

5 Conclusions and perspectives

In this paper, we presented our anomaly detection approach, which is inspired by emotional processes, for improving the resilience of cyber-physical systems. We discussed some resilience definitions as well as our position in relation to resilience-related approaches.

In order to detect abnormal situations, we integrate individual and collective emotion-inspired processes into an agent architecture. For detecting abnormal situations, our approach includes a statistical model, a perception grid, an

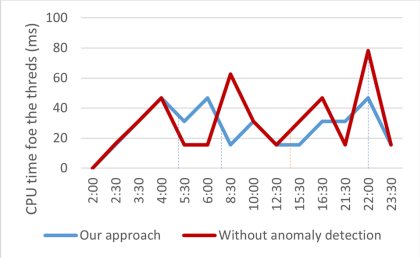

Fig. 12. CPU time using scenario 1.

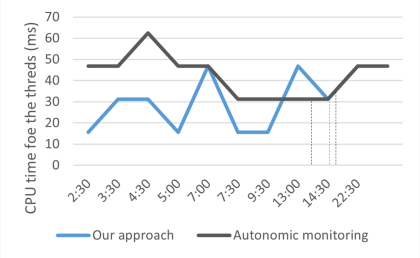

Fig. 13. CPU time using scenario 2.

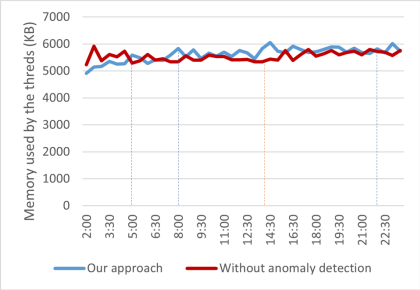

Fig. 14. Used memory (RAM) using scenario 1.

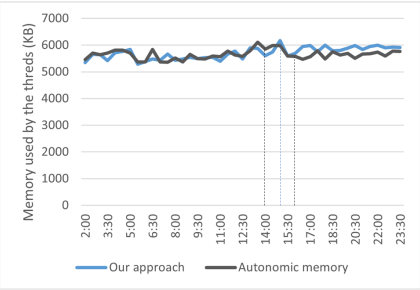

Fig. 15. Used memory (RAM) using scenario 2.

episodic memory, and a collective detection mechanism. Individual detection processes have an impact on the system's control group, triggering a collective detection to affirm or reject a suspicious situation. The agent's knowledge (perception grid and episodic memory) and behavior are affected by a confirmed situation (sampling frequency and tolerance thresholds). This architecture also includes processes for adapting individual and collective behavior in response to detected situations in order to improve the resilience of CPSs.

The detection approach and the R-ECM architecture were implemented and validated by simulating two scenarios. The results were interesting, and we hope to obtain more relevant results in the future. We plan to apply our approach to other systems with more complex scenarios.

References

1. Anderson, J.R.: Cognitive psychology and its implications. Worth publishers (2000)
2. Bißmeyer, N., Mauthofer, S., Bayarou, K.M., Kargl, F.: Assessment of node trustworthiness in vanets using data plausibility checks with particle filters. In: 2012 IEEE Vehicular Networking Conference (VNC). pp. 78–85. IEEE (2012)
3. Bordini, R.H., El Fallah Seghrouchni, A., Hindriks, K., Logan, B., Ricci, A.: Agent programming in the cognitive era. Autonomous Agents and Multi-Agent Systems **34**, 1–31 (2020)

4. Bourgais, M., Taillandier, P., Vercouter, L., Adam, C.: Emotion modeling in social simulation: a survey. Journal of Artificial Societies and Social Simulation (2018)
5. Brooks, R.: A robust layered control system for a mobile robot. IEEE journal on robotics and automation **2**(1), 14–23 (1986)
6. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM computing surveys (CSUR) **41**(3), 1–58 (2009)
7. Colabianchi, S., Costantino, F., Di Gravio, G., Nonino, F., Patriarca, R.: Discussing resilience in the context of cyber physical systems. Computers & Industrial Engineering p. 107534 (2021)
8. David, O.E., Netanyahu, N.S.: Deepsign: Deep learning for automatic malware signature generation and classification. In: 2015 International Joint Conference on Neural Networks (IJCNN). pp. 1–8. IEEE (2015)
9. Falcone, Y., Mariani, L., Rollet, A., Saha, S.: Runtime failure prevention and reaction. In: Lectures on Runtime Verification, pp. 103–134. Springer (2018)
10. Fauri, D., Dos Santos, D.R., Costante, E., den Hartog, J., Etalle, S., Tonetta, S.: From system specification to anomaly detection (and back). In: Proceedings of the 2017 Workshop on Cyber-Physical Systems Security and PrivaCy. pp. 13–24 (2017)
11. Jones, A., Kong, Z., Belta, C.: Anomaly detection in cyber-physical systems: A formal methods approach. In: 53rd IEEE Conference on Decision and Control. pp. 848–853. IEEE (2014)
12. Kouicem, E., Raïevsky, C., Occello, M.: Towards a Cyber-physical Systems Resilience Approach based on Artificial Emotions and Multi-agent Systems. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence (ICAART 2020) - Volume 1. pp. 327–334 (2020)
13. Kouicem, E., Raïevsky, C., Occello, M.: Emotional processes for cyber-physical systems resilience. In: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 333–338. IEEE (2021)
14. Linkov, I., Kott, A.: Fundamental concepts of cyber resilience: Introduction and overview. In: Cyber resilience of systems and networks, pp. 1–25. Springer (2019)
15. Malhotra, P., Vig, L., Shroff, G., Agarwal, P., et al.: Long short term memory networks for anomaly detection in time series. In: Proceedings. vol. 89, pp. 89–94 (2015)
16. Müller, J.P., Pischel, M.: The agent architecture interrap: Concept and application (1993)
17. Ratasich, D., Khalid, F., Geissler, F., Grosu, R., Shafique, M., Bartocci, E.: A roadmap toward the resilient internet of things for cyber-physical systems. IEEE Access **7**, 13260–13283 (2019)
18. Scherer, K.R., Schorr, A., Johnstone, T.: Appraisal processes in emotion: Theory, methods, research. Oxford University Press (2001)
19. Sisto, A., Vicinanza, F., Campanozzi, L.L., Ricci, G., Tartaglini, D., Tambone, V.: Towards a transversal definition of psychological resilience: a literature review. Medicina **55**(11), 745 (2019)
20. Trivedi, K.S., Kim, D.S., Ghosh, R.: Resilience in computer systems and networks. In: Proceedings of the 2009 International Conference on Computer-Aided Design. pp. 74–77. ACM (2009)
21. Tulving, E., et al.: Episodic and semantic memory. Organization of memory **1**, 381–403 (1972)
22. Woods, D.D.: Essential characteristics of resilience. In: Resilience engineering, pp. 21–34. CRC Press (2017)