Merging Digital Twins and Multi-Agent Systems Approaches for Security Monitoring

Zoé Lagache[†], Annabelle Mercier[*] , Oum-El-Kheir Aktouf[*] and Arthur Baudet[*]
[*]Univ. Grenoble Alpes, Grenoble INP, LCIS, 26000 Valence, France
e-mail: {firstname.lastname}@lcis.grenoble-alpes.fr
[†]Univ. Grenoble Alpes, CEA, Grenoble, France zoe.lagache@cea.fr

***Abstract*—This paper proposes a method for designing a model based on the Multi-Agent System (MAS) and Digital Twin (DT) concepts to study the cyber-physical systems security. When Cyber-Physical Systems (CPS) are used in a network to address a complex problem (such as the deployment of smart cities, Industry 4.0, etc.), they present a unique wide vulnerability challenges as their attack surface ranges from hardware and physical attacks to software attacks and even including network attacks. To meet these challenges, we explored several approaches to ally MAS and DT with the aim to benefit from the scalability and adaptability of MASs and the enhanced modelling of DTs. As a result of this exploration, we present a novel approach to tackle networking attacks of CPSs. To showcase our approach, we present its application to detect blackhole attacks (a kind of attack in which one or more nodes attract all communications and not forward them, mimicking an error in the network) in a simulated smart home environment. As results are promising, we conclude and discuss future research perspectives in allying DTs and MASs for managing the security of CPSs.**

***Keywords*- Multi-Agent Systems, Digital Twins, Cyber Physical Systems, Network Security**

I. Introduction

Cyber-Physical Systems (CPSs), defined as the interaction between physical systems and processes using computations and communication abilities with the Cyber-Physical Systems Steering Group [1], can be found everywhere: in vehicles to control safety mechanisms such as airbags and belt tensioners, to monitor in manufacturing plants [2] or acting as sensors and actuators components of smart homes and smart cities [3]. However, these ubiquitous systems can also be vectors of attacks, such as the unauthorized access, manipulation of system controls, and the disruption of critical infrastructure. These risks can have significant consequences, including loss of life, economic damage, and the disclosure of sensitive information [4] [5]. It is important to have robust security measures in place to mitigate these risks and to have available contingency plans to respond to potential security breaches. CPSs can be found in a wide range of applications and could benefit from a stronger degree of security. Our approach, motivated by this need of security, aims at contributing to CPSs security field by combining innovative approaches such as, Multi-Agent System (MAS) and Digital Twin (DT) models.

A MAS is a system composed of agents collaborating with each other in order to achieve a common goal. These agents communicate with their local neighbors and typically possess only a limited and localized view of the overall system. MAS are now a trend in the Internet of Things (IoT) field thanks to its decentralization aspect. Furthermore, a DT is often associated with a way to track and analyze a system in real time, usually in order to predict its behavior. Both MAS and DT hold very useful potential for modeling CPS. These two concepts are complementary in modeling, securing, and preventing cyberattacks on CPS, DT allows for dynamic simulations of system behavior under various scenarios, while MAS provides a deeper understanding of complex interactions between system components. When combined, these models provide a more comprehensive approach to identifying vulnerabilities, testing security measures, and optimizing system design to enhance security. Our contribution introduces a novel method for modeling complex CPSs, facilitating the identification of potential security vulnerabilities and the development of strategies to improve security and protect against cyber threats.

Section II proposes a brief related work. Section III presents the motivations for our study and the main points about CPS vulnerabilities. In Section IV, we introduce the different possibilities for leveraging the DT and MAS models for monitoring CPS and detecting vulnerabilities, and explain the adopted model. In Section IV-D, we discuss the application of our model to the case study of a blackhole attack detection in a smart home setting. Validation of this application is then presented in Section V. Finally, we conclude in Section VI.

II. Related work

DTs are often used in the field of complex system monitoring. For example, they can be used to simulate product quality in a manufacturing process [6], incorporating real-time data from IoT sensors to improve simulation accuracy and reduce uncertainty. As for MASs, they can model distributed and heterogeneous systems [7] which makes them also good candidate for simulating CPS. In terms of security analyses, a framework for evaluating the security of a system is named ”cia model” [8] which uses three main criteria to assess the overall security: *Confidentiality* refers to the protection of sensitive information from unauthorized access or disclosure ; *Integrity* refers to the protection of information from unauthorized modification or destruction ; *Availability* refers to the ability of authorized individuals to access the information when they need it. The association of DT and MAS for security is a topic that is not much studied in literature, as evidenced by the scarcity of relevant studies or papers. The available literature is also often highly specialized, making it difficult to find comprehensive information on the topic. For example, work in [9] focuses on the medical field and present

ISBN: 978-1-68558-206-7

an agent-based DT to advise severe traumas, which is their use case. Other examples exist, such as the work in [10] [11] that focus on smart cities and farms management, respectively. Another example is [12] which discuss the existing literature about the extended reality in systems such as CPS, but it does not give any insights on security of such systems. Nevertheless, we can encounter some more general papers in very recent works [13], which is a review on MASs in support of DTs. They present the main challenges like a roadmap for DT and MAS in general context, but this paper does not focus on security issues, and [14] that explore the development of Artificial Intelligence in digital ecosystems and focus on how to make them safer. However, this last paper does not explicitly refer to MAS but to collaborative systems. The work in [15] presents a generic way of modelling DT using the MAS idea, discusses the difficulty in building DTs and creates a method to make this goal easier.

III. CPS VULNERABILITIES

A. Our CPS model

As explained in the introduction, CPSs are the combination of the physical world and the cyberspace interacting with one another through the use of sensors, actuators, communication, and interfaces. The cooperation between the physical and cyber systems is typically achieved through the use of sensors to monitor the physical system and actuators to control it, as seen in [12] [16] [17]. Monitoring refers to the process of gathering data and information about a system, process or environment, by using sensors and virtual models. Figure 1 represents an abstract view of a CPS. We simplified a CPS as two parts: the Physical Process(es) (PP) part and the cybersystem part. Both of these parts are interacting with each other through sensors and actuators, which compose the interface between both worlds. The Cyber System (CS) is composed of computing devices receiving data from the sensors, processing it, and sending the result to the actuators. The green arrows indicate the monitoring interaction, while the red ones represent the communication within the CS. The numbers point to the parts of the CPS that are subject to vulnerabilities.

B. Identification of vulnerabilities

By analyzing and synthesizing the classifications done in [4] [5] [18], we define four attacks classes shown in Table I. First, communication attacks have the potential to be operated on all communication links, i.e., points ②, ④, ⑥ and ⑦. For example, *eavesdropping* is a passive attack where the attacker is listening to a communication between two or more nodes and (Confidentiality criteria is affected) in the *Man-In-The-Middle (MITM)*, the attacker can intercept the communication packets and thus to tamper with them (Confidentiality and Integrity). Second, network or routing attacks are attacks that are the result of a changing behavior from one element of the system that can impact changes in the rest of the network. All parts of the system from ① to ⑤ could be impacted by such attacks. In a *blackhole attack*, the attacker is able to corrupt one or more nodes in the networked system to make them advertise fake routes that are shorter than those of its neighbors. However, once the blackhole nodes receive a packet, they drop it and Availabity is affected. Other attacks in this class are *greyhole attack* and *wormhole attack*. Third, physical attacks can be done on the devices that are the closest to the real world, thus on the actuators or on the sensors (① and ⑤). They can be *side channel attack*, *fault injection attack* where Integrity is affected and all criteria for *jamming attack*. Fourth, miscellaneous attacks have their own locations on the map and are not part of previous classes and could affect all criteria. For example, *a malware spreading attack* is an application attack where the attacker spreads a piece of malicious code into one or more computing devices (③). Another attack can be a *DoS attack* where the attacker disables a device, so it cannot work anymore, which can be located either on point ② or on point ③ and affects availability.

TABLE I
SUMMARY OF CPS VULNERABILITIES.

Attack Class	Attack	Vulnerable Surface	CIA Involved
Communication	Eavesdropping	②, ④, ⑥ & ⑦	Confidentiality
	MITM		Confidentiality, Integrity
Network/Routing	Blackhole	①–⑤	Availability
	Greyhole		Availability
	Wormhole		All
Physical	Side Channel	①,⑤	Confidentiality
	Fault injection		Confidentiality
	Jamming		Availability
Miscellaneous	Malware	③	All
	DOS	② or ③	Availability

Most of these attacks can be avoided with preventive methods. Eavesdropping and MITM attacks can be prevented by encrypting the communications, the side channel and fault injection ones by the specific algorithms or Trusted Execution Environments, and malwares by computing and comparing checksums of binaries. That is why our work is focused on the detection of routing attacks. More precisely, we chose to work on blackholes detection in Section V because they exist in less diverse and elaborated versions than the other attacks.

IV. A MODEL TO MERGE DT AND MAS FOR MONITORING SECURITY IN CPS

As there are few studies that focus on MAS and DTs at the same time, and even fewer that are concerned with system security, we propose here a study of the different possibilities to compose the MAS and DT models for a scalable solution to manage security in CPSs. In this section, we present different approaches to leverage MAS and DTs solution to secure CPS,

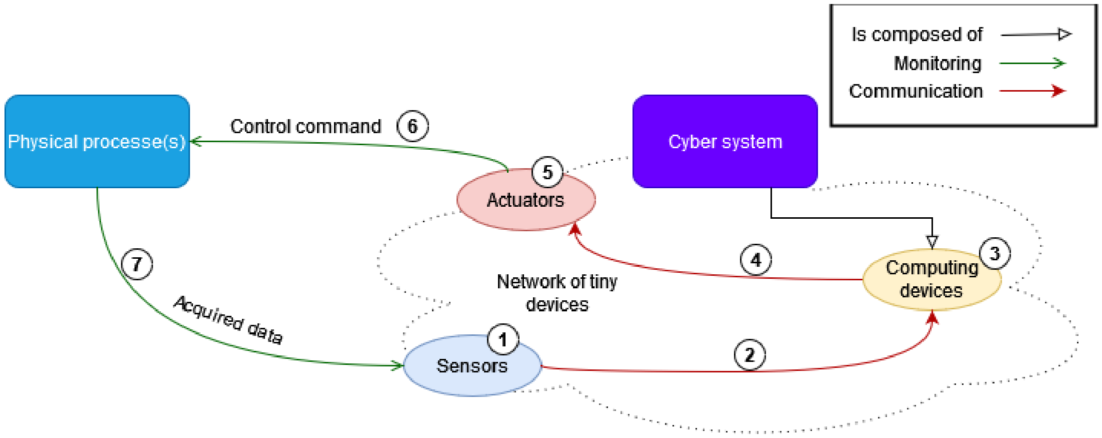


Figure 1. CPS main vulnerabilities

compare them and then present the approach we adopted and implemented, for the validation in the rest of the paper. To understand the models that we propose and to compare them, we will first explain the model architecture elements, and then we will introduce a guiding example to follow the way and processing of a measurement returned by a sensor.

The **Physical system Process(es)** (PP) is composed of sensors and actuators. For example, a sensor measures a room's temperature and an actuator increases or decreases the radiator thermostat. The physical system represents either a single element or is composite by bringing together several objects. The **Cyber System** (CS) is the digital entry point of the PP one. It has a module that receives datas, a module that transmits actions, and a module that processes the information. The CS is configured based on the hardware's organization of the constituting the physical system. We call the combination of CS and PP a Cyber-Physical System (CPS). We define a digital twin as a system that digitises the process and data flow of the physical system, with a feedback loop injecting new information via the cyber system to monitor or improve the physical system. In our study, the digital twin does not possess intelligence, it only digitizes information. The intelligence of the system to process data and audit potential security flaws lies at the agent level. The multi-agent system composed of several agents (upper than 3 agents) constitutes an intelligent virtualization of the system. According to the presented architectures, the physical system consists of sensors and actuators, and the cyber system consists of processing units. To understand the architecture of the proposed models, we use a guiding example with the following nominal scenario. For each model, we will explain the way of the sensored measurement, information processing, and subsequent actions.

In the context of smart home technology, a room is equipped with a physical system consisting of temperature sensors and actuators that can order the adjustment of the thermostat of a heating device or actuator to control the opening of the shutters in a room. The nominal scenario consists of regulating the temperature to maintain 20°C in a given space (room, house). The CPS returns 19°C via a temperature sensor, and the multi-agent system processes the information through the agent associated with the sensor. Depending on the behavior of the agent, an order may be directly sent to the actuator to open the shutters or increase the thermostat of the heating system. However, a more complex behavior involving information exchange in the system can be adopted. In this case, monitoring the security of the system becomes important. The agent may request measurements from its neighbors to obtain additional temperature readings from the physical system and wait for this new data to make a decision. For example, if the neighboring measurements are below 20°C, it could decide to send an order to increase the thermostat of the heating system.

A. A MAS Composed of Digital Twins of Cyber-Physical Systems

In this first approach, we consider a set of CPSs communicating and interacting with each other. Each CPS run DTs of its physical processes, and each CPS is considered an agent of global MAS. This idea is illustrated in Figure 2. Such an approach would allow the CPSs to produce a complete view of their processes thanks to the DTs models and use MAS capabilities to organize and handle the information in an autonomous and dynamic fashion. An example of the application of this approach, as a security framework, would be to deploy embedded systems, monitors, running DTs of sensors and actuators, sharing the information with nearby CPS to compare values to detect attacks on the sensed values. This model is used to manage a set of DT/CS. An agent represents the DT/CS pair and retrieves the sensors' measurements and executes the orders transmitted to the actuators. In our example, a sensor measures 19°C, the DT digitises the information and the associated agent processes the datas, for example by sending measurement requests to neighbouring agents. Once the information has been returned, a decision can be made to wait if the values returned are above 20°C or to ask the CS to send a command to increase the heating thermostat.

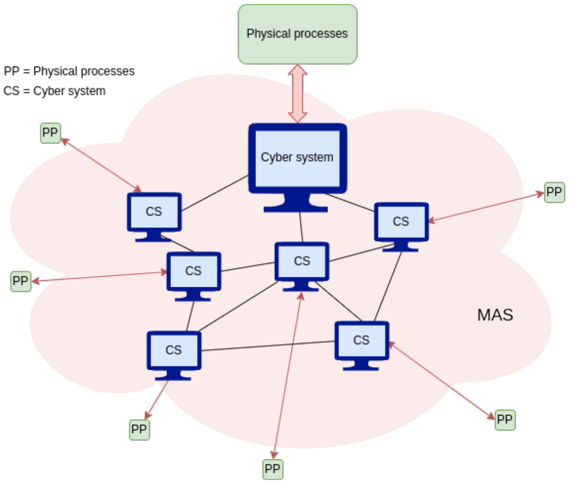


Figure 2. MAS composed of CPSs running Digital Twins of their processes.

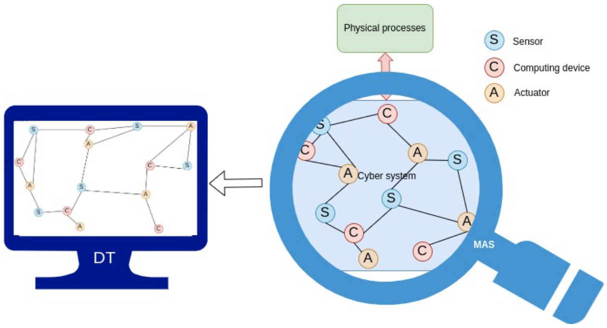


Figure 3. Digital Twin of the CPSs controlled by a Multi-Agent System.

In this approach, the DTs enable a better modelling of the physical processes, including computed information, to provide a better viewpoint of processes handled by the monitors. To not merge all information of the DTs but rather use a MAS approach provides a better scalability as well as better behaviors under changes of the systems: the monitors will first try to coordinate with their neighbors, without overloading the whole systems with a large amount of data, and also be able to re-organize at run-time if a monitor stops working or if new ones are added (including different ones since MASs allow for the cooperation of heterogeneous systems).

The main drawback of this approach is that it creates a “system of systems” which itself brings its one challenge in terms of security [19]. Since the monitors communicate with each other’s, their communications or even themselves can be attacked. Moreover, such systems are also difficult to be modelled since they rely on the coordination of multiple systems, which can create numerous possible interactions and states in which the whole system can be.

B. Multi-Agent Algorithm Monitoring a Digital Twins of CPSs

The second approach also focuses on a network of CPSs, but this time, they are only communicating with their respective physical processes and a server, in charge of analyzing the information of the whole system. This server is running DTs for each CPS and a MA algorithm to analyze the modelled system. The output of the MA algorithm is then used to send control commands to the CPS through their DTs. This approach is illustrated in Figure 3.

Such an approach would provide a single, central view of the whole system thanks to the DTs. Unlike the previous approach, the MAS approach does not enable decentralized control but rather enable a more coherent way of analyzing the network of DTs, a distributed system (for example, by attributing an agent to each DT). An example of the application of this approach would be an industry 4.0 plant in which each component (robots, packages) is modelled with a DT and the MA algorithm provides on-line analysis of the DTs to detect incoherent behaviors due to attacks on the production lines [20].

In this model, an agent is associated with the sensors and actuators of the PP and the computing device of the CS. All the sensors, actuators, and processing unit make up a single CPS. The DT provides a snapshot of the system. In our scenario, the sensor reads 19°C, the processing unit and its associated agent process the information and can instruct the actuator to raise the temperature or ask its neighbours for other readings. Here, the DT has a supervisory role; for example, if the system is overloaded or if the data seems inconsistent, the operator, who can see what is happening, can send instructions directly to the CPS system.

This approach is the canonical use of DTs and CPS and benefits from the advantages of DTs, an enhanced view of the whole system execution for an optimal control and CPSs analysis. The MAS provides a natural way of designing the analysis and control algorithms as well as enabling horizontal scaling, some of the agents can be located on different servers to scale to larger numbers of CPSs. Moreover, since the control is centralized, it is possible for an operator to get a global view of the system. However, since it is centralized, the server running the DTs becomes a Single-Point-Of-Failure (SPOF) and needs to be hardened since it will be a target of choice for an attacker. If the server is compromised or its communications prevented, the CPSs will no longer receive commands and attacks will remain undetected due to the lack of information provided by the CPSs.

C. Digital Twin of the Cyber-Physical System seen as a MAS

Unlike the two previous one, the third approach focuses on a single CPS. In this approach, the DTs modelled the sensors and actuators of the CPS and, like the second approach, a multiagent (MA) algorithm is used for analyzing the behavior of the DTs. This approach is illustrated in Figure 4.

This third approach allows for an analysis at the sensors and actuators level, as the first approach, with a central control, as the second approach. An example of the application of this approach could be the fine-grained detection of attacks

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-206-7

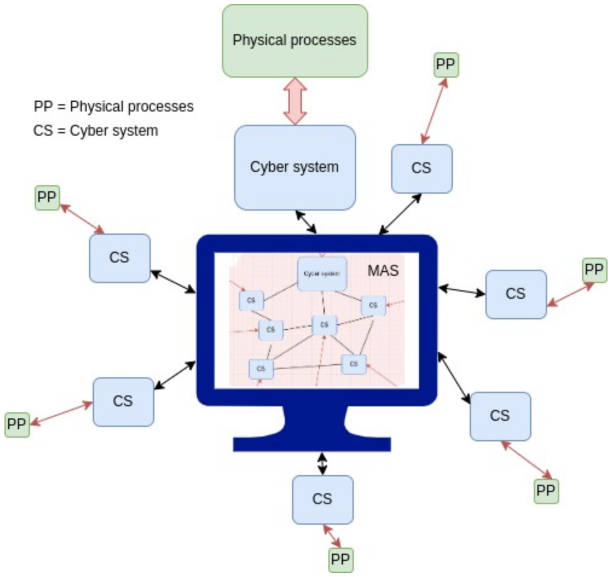


Figure 4. MA algorithm monitoring DTs of the components of a CPS.

in one production line by detecting incoherent values given by the DTs of the sensors and behaviors of the DTs of the actuators, each one an agent of a MAS. In this model, the cyber system and the physical process are able to receive, transmit and process the datas, they form a single CPS. An agent is associated at each CS to a server. The multiagent system on the server is the brain of the system, we obtain the digitalisation of the whole elements of the CPS. The sensored data is sent to the associated agent on the server. The multiagent makes decisions and will transmit the actions to the CPS.

This approach provides an interesting trade-off between the centralization of all the data of a whole network and the complexity of controlling a decentralized system. The analysis is done near at the edge of the network, on each CPS, thanks to the modeling capabilities of the DTs as well as the efficiency of the multi-agent paradigm. While, it may not be possible to run a machine learning and deep learning algorithm on the CPS, a multiagent algorithm may detect incoherent behaviors at a lower computational and energetic cost.

However, this approach only focuses on one CPS at the time, which may not be enough to detect large scale attacks spanning over a whole plant, smart home or network of vehicles. It also does not provide any solution if the whole CPS (and not only the actuators and sensors) is under attack. To summarize, Table II gives a comparison with advantages and drawbacks of the three models.

D. Chosen model for monitoring security issues in CPSs

We propose a new approach which is a compromise to avoid the pitfalls of the systems-of-systems model, while leveraging MAS benefits. We propose a multi-agent algorithm used to analyze information of the DTs of the components of a single CPS, but not running on the CPS itself, but rather on a distant server. This approach is illustrated in Figure 5. It is a trade-off between the three approaches presented above and is a good candidate for a proof-of-concept as it encompasses most of the notions of the approaches while remaining simple enough to be properly validated.

This approach could be used in a smart home system setup: the CPS would be composed of the actuators (heating, lights, kitchen appliances, etc.) and sensors (presence or smoke detector, light sensor, etc.). Each CPS component would send information to a central server, which would then use DTs to model the CPS and a MA algorithm to analyze the behaviors of the components in order to detect attacks on the CPS.

The use of a central server creates a SPOF, which is a major drawback but which also drastically decrease the difficulty of deploying our approach: the MA algorithm has access to all the information on a single device and does not have to rely on coordination between the CPS components to take a decision. Moreover, it is easier to harden the security of the server rather than to each component of the CPS. The DTs can also serve as a monitoring tool for users and operators and can be used to ease the CPS maintenance.

The server that digitises the information constitutes the physical system's DT. Here, we add the notion of sensor twins, which are visualised on the DT diagram. The agents associated with the sensor twins make up the SMA: the granularity can be chosen to process the data in the agents. Thermometers can be combined, humidity sensors or both. Sensors could also be combined by room type (neighbouring, north, south). In this way, the multi-agent system could adopt different processing rules depending on the location of the building or its function. Once a measurement of 19° has been taken, a notification is sent to the DT. The sensor twin updates T°=19. The agent retrieves this information, and can apply its process and decision rules by asking neighbouring agents for the temperature they have recorded, and make the same type of decision as in the previous models.

This approach also works under the assumption that some components can reliably send information to the server. If some components do not send information to the server, we expect that the MA algorithm will detect their absence and adapt accordingly (e.g., raising an alarm, and re-organizing if the loss was expected).

V. Proof of concept: blackhole detection

We have done a general simulation with Mesa framework [21] to simulate communication in a CPS and to test our architecture on a blackhole detection. The source code of the experiments is available on GitHub [22]. As explained in Section III-B, we consider the blackhole attack because it can be detected by a behavior analysis that suits well with the use of a MAS. Only attacks directed toward the CPS will be taken into account. Attacks on the server part in Figure 5, or its communication with the CPS are not investigated. To lead our experiments, we make several hypotheses about the

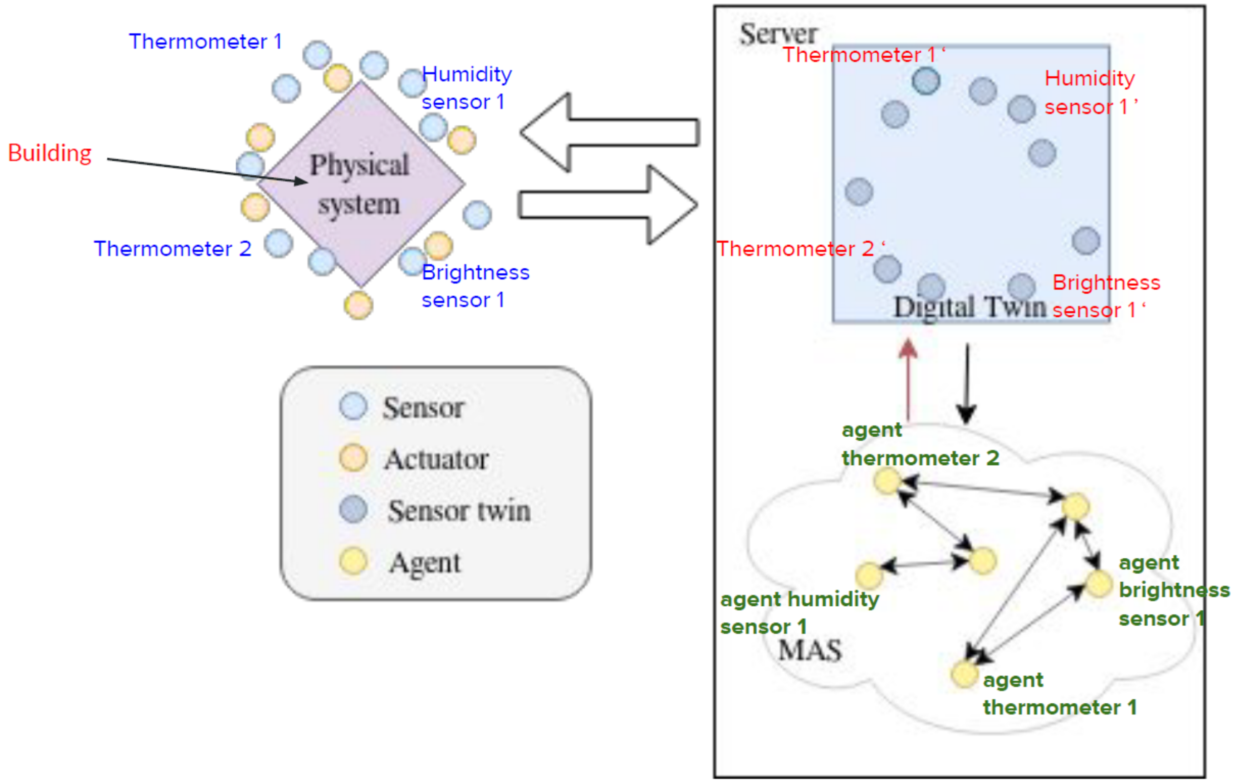


Figure 5. Chosen model: Monitoring the DTs of a CPS using a multi-agent algorithm.

TABLE II
COMPARISON OF THE THREE PROPOSED APPROACHES.

No.	Benefits	Drawbacks
1	• High scalability • High adaptability at run-time	• Hard to maintain • Only provides a global security analysis • Creates new vulnerabilities
2	• Enables part of MAS scalability without the difficulties of the system-of-systems approach • Provide a global view of the system	• Only provide a global security analysis • The server is a single-point-of-failure
3	• Low impact on the system as it runs directly on the CPSs • Highly scalable	• Only provide a local security analysis • No redundancies, if one CPS is under attack, the others will not be able to detect it

attacker's goals, capabilities, and knowledge. The goal of the attacker, in our chosen scenario, is to achieve a blackhole on the CPS network. Thus, to alter the availability of the system through the usage of corrupted nodes that will drop packets and potentially advertise maliciously. Thus, the attacker can add intruder nodes as well as corrupt a victim node. Since some metadata on exchanged packets, such as the source and destination addresses, are important to analyze them. For the experiments, we simulate the CPS by a network of nodes that exchange messages. The requirements on the CPS structure are defined as follows: (1) it is made of sensor nodes communicating with each other forming a network, (2) nodes can be added or removed from the network, and (3) a node cooperates with the DT by sending it notifications. The considered messages to help the blackhole algorithm detection are: (i) `data` messages: messages containing raw datas used by the higher level application, (ii) `advertisement` messages: messages to determine which path must be used between two nodes. In most protocols, the value of this type of message starts from 0 from the sending node and is incremented each time a new node receives it, so the sending node knows its closest neighbors. In order to detect the blackhole of communication between nodes, we also assumed that sensor nodes are sending notification to the server, giving information on messages received and sent. If a node does not cooperate, it is either because it is not part of our system or because it is malicious. In the second case, either the node does not communicate with other nodes of the system, and thus is not harmful to it, or it communicates and will therefore be seen through the notifications from other nodes. The implementation process is defined as follows. The DT block receives the inputs, which are the sensor nodes notifications. Since our tests do not work in real time, a YAML is adopted to simulate data exchange between nodes. A message has a source ID and a destination ID and information on message (message type, datas and

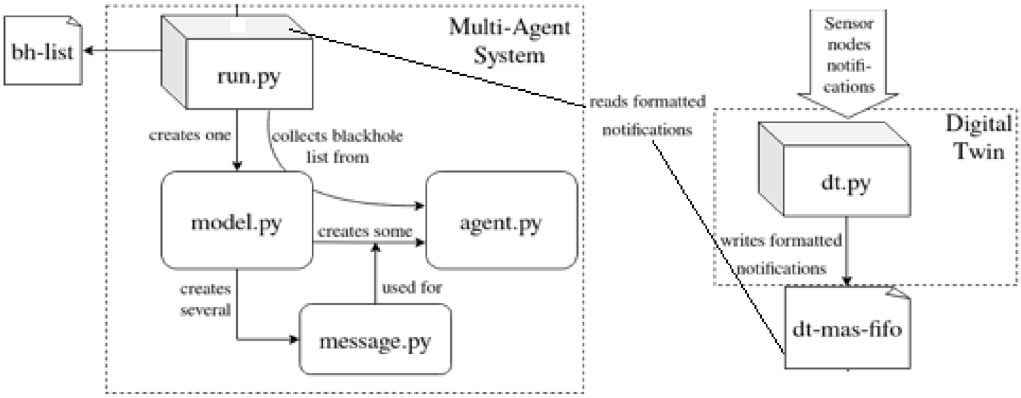


Figure 6. Implementation architecture with Mesa.

source if of previous node) ; when the message could not be sent directly there is the destination ID of the next node and ; when the message can not be sent directly the source ID and destination ID of second message.

Each of the test files contains each notification received by the DT from each node. Thus, they contain all messages sent and received by each sensor. The dt.py python code writes these notifications into a FIFO. On the other side of this FIFO, the run.py python code, which is the entry point of the MAS block, reads the notifications and launches Mesa. The Mesa agents analyze the data and create a blackholes list, which is retrieved by run.py. More precisely, the run.py python code reads the notifications from the FIFO and creates a Mesa model which is our MAS model. The model.py python code is what creates and keeps updated the Mesa agents and messages object, which are respectively described in agent.py and message.py. During running time, Mesa agents will analyze the messages they have and create a list of tags, or states, indicating which node they consider blackhole from their local point of view. Mesa includes a time system based on steps. We write what the agents and the model do each step. Each agent analyses itself and the nodes from which it receives messages at each step. The architecture implementation is shown in Figure 6. The agent behavior checks the three elements to allow the blackhole detection: (1) whether the number of messages of type Data it sends is above the threshold given by the user, (2) whether it forwards messages received that are not destined to it, and, (3) whether it receives advertisement messages of value 0. Indeed, a lot of network protocols use a system of advertisement messages to determine which path must be used between two nodes. The value of this type of message is incremented each time a new node receives it. Thus, it is not possible for a node to receive advertisement messages of value 0. The blackholes lists are then retrieved and merged with the Mesa Model at each step before being put into a file by run.py.

Several tests are leading to evaluate the effectiveness of the proposed solution. We created some data sets in which we deliberately included anomalies, to be able to check the effectiveness of our code. Each of these tests are stored in a YAML file which is read in one go when executed. We made a total of 9 test files, checking if the 3 analysis on a basic configuration network of nodes Figure 7.

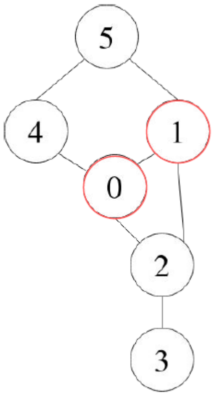
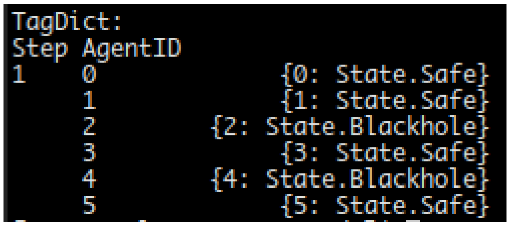


Figure 7. Test configuration network.



```
TagDict:
Step AgentID
1    0                {0: State.Safe}
     1                {1: State.Safe}
     2           {2: State.Blackhole}
     3                {3: State.Safe}
     4           {4: State.Blackhole}
     5                {5: State.Safe}
```

Figure 8. Detection of blackhole.

First, we make one test without any anomaly, to check that nothing is detected in this case. To complete simple cases, we make three other tests to check if a blackhole is detected if (1) a node does not forward a message (2) a node does not send any Data type messages; and (3) a node sending a wrong advertisement. Then, tests are leading to check that detecting a malicious node does not impact the detection of another one in three other cases (1) a node not forwarding and another not sending Data messages; (2) a node not forwarding and another sending wrong advertisement; and (3) a node not sending

Data messages and another sending a wrong advertisement. The last tests have to check that having multiple anomalies in one node does not impact the blackhole detection with the following situations: (1) a node not sending Data messages and sending a wrong advertisement; and (2) a node not forwarding and sending a wrong advertisement. The scenario of testing a situation with two suspicious nodes is as follows: Node 2 does not send any messages except its advertisement. Node 4 does not forward one (or more) message it received that was destined to another node. It is expected that Node 4 detects itself as a blackhole because it did not forward one message or more, and node 2 detects itself because it did not send any Data messages. The obtained results are shown in Figure 8: node 2 detecting that it is not sending any messages, while node 4 detects that it is not forwarding one (or more) message. Thus, they changed their own tag for the blackhole tag.

VI. Conclusion

Our initial investigations revealed that digital twins and multiagent systems for security issues in CPS are rarely explored together, underscoring the need for further research in this area. After analyzing CPS vulnerabilities and outlining MAS and DT, we considered various approaches to integrate both concepts into a cohesive model. The final model was developed by selecting the most relevant components, aligned with our objectives and constraints. To validate our approach, we conducted an experimental implementation using the Mesa framework simulation tool. This implementation featured a basic algorithm for detecting blackholes in a Wireless Sensor Network (WSN)-like environment, along with a preliminary set of tests to evaluate its functionality. However, the lack of access to a real CPS or a comprehensive dataset posed significant limitations, leaving our analysis incomplete. Given these constraints, particularly the absence of benchmarks and an actual CPS, the most promising future work would involve implementing our model on a real CPS capable of sending real-time notifications. This would allow for the collection of realistic data and enable the evaluation of the model's performance, as well as the potential integration of feedback mechanisms into the CPS.

References

[1] CPS Steering Group, "Cyber-Physical Systems Executive Summary," Chicago, IL, USA., 2008.

[2] B. Dafflon, N. Moalla, and Y. Ouzrout, "The challenges, approaches, and used techniques of cps for manufacturing in industry 4.0: a literature review," *The International Journal of Advanced Manufacturing Technology*, vol. 113, no. 7, pp. 2395–2412, 2021.

[3] C.-S. Shih, J.-J. Chou, N. Reijers, and T.-W. Kuo, "Designing cps/iot applications for smart buildings and cities," *IET Cyber-Physical Systems: Theory & Applications*, vol. 1, no. 1, pp. 3–12, 2016.

[4] E. K. Wang, Y. Ye, X. Xu, S. M. Yiu, L. C. K. Hui, and K. P. Chow, "Security Issues and Challenges for Cyber Physical System," in *2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, 2010, pp. 733–738.

[5] S. Singh, N. Yadav, and P. K. Chuarasia, "A Review on Cyber Physical System Attacks: Issues and Challenges," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, 2020, pp. 1133–1138.

[6] H. Marah and M. Challenger, "Intelligent agents and multi agent systems for modeling smart digital twins," *Engineering multi-agent systems*, 2022.

[7] M. E. Gregori, J. P. Cámara, and G. A. Bada, "A jabber-based multi-agent system platform," in *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '06, 2006, p. 1282–1284.

[8] NIST, "Security and privacy controls for information systems and organizations," 2020.

[9] A. Croatti, M. Gabellini, S. Montagna, and A. Ricci, "On the integration of agents and digital twins in healthcare," *Journal of Medical Systems*, 2020.

[10] T. Clemen, N. Ahmady-Moghaddam, U. A. Lenfers, F. Ocker, D. Osterholz, J. Ströbele, and D. Glake, "Multi-Agent Systems and Digital Twins for Smarter Cities," in *Proceedings of the 2021 ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*, 2021, pp. 45–55.

[11] V. Laryukhin, P. Skobelev, O. Lakhin, S. Grachev, V. Yalovenko, and O. Yalovenko, "Towards developing a cyber-physical multi-agent system for managing precise farms with digital twins of plants," *Cybernetics and Physics*, pp. 257–261, 2019.

[12] Y. P. Tsang, T. Yang, Z. S. Chen, C. H. Wu, and K. H. Tan, "How is extended reality bridging human and cyber-physical systems in the IoT-empowered logistics and supply chain management?" *Internet of Things*, vol. 20, p. 100623, 2022.

[13] E. Pretel, E. Navarro, V. López-Jaquero, A. Moya, and P. González, "Multi-Agent Systems in Support of Digital Twins: A Survey," in *Bio-inspired Systems and Applications: from Robotics to Ambient Intelligence*, 2022, pp. 524–533.

[14] E. Cioroaica, B. Buhnova, and E. Tomur, "A paradigm for safe adaptation of collaborating robots," in *Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, 2022, pp. 113–119.

[15] H. Marah and M. Challenger, "Madtwin: a framework for multi-agent digital twin development: smart warehouse case study," *Annals of Mathematics and Artificial Intelligence*, vol. 92, pp. 1573–7470, 2023.

[16] L. Chen, F. Hu, S. Wang, and J. Chen, "Cyber-physical system fusion modeling and robustness evaluation," *Electric Power Systems Research*, vol. 213, p. 108654, 2022.

[17] C.-U. Lei, K. Wan, and K. L. Man, "Developing a Smart Learning Environment in Universities Via Cyber-Physical Systems," *Procedia Computer Science*, vol. 17, pp. 583–585, 2013.

[18] M. Wazid, A. K. Das, S. Kumari, and M. K. Khan, "Design of sinkhole node detection mechanism for hierarchical wireless sensor networks," *Security and Communication Networks*, vol. 9, no. 17, pp. 4596–4614, 2016.

[19] A. Baudet, O.-E.-K. Aktouf, A. Mercier, and P. Elbaz-Vincent, "Systematic Mapping Study of Security in Multi-Embedded-Agent Systems," *IEEE Access*, vol. 9, pp. 154 902–154 913, 2021.

[20] T. Horak, P. Strelec, L. Huraj, P. Tanuska, A. Vaclavova, and M. Kebisek, "The vulnerability of the production line using industrial iot systems under ddos attack," *Electronics*, vol. 10, no. 4, 2021.

[21] J. Kazil, D. Masad, and A. Crooks, "Utilizing python for agent-based modeling: The mesa framework," in *Social, Cultural, and Behavioral Modeling*, R. Thomson, H. Bisgin, C. Dancy, A. Hyder, and M. Hussain, Eds., vol. 12268, 2020, pp. 308–317.

[22] Z. Lagache, A. Baudet, A. Mercier, and O.-E.-K. Aktouf, "Ghithub link for experimental." [Online]. Available: https://hal.science/hal-04720424