international journal of critical infrastructure protection 53 (2026) 100846

Contents lists available at ScienceDirect

International Journal of Critical Infrastructure Protection

journal homepage: www.elsevier.com/locate/ijcip

Situational awareness for trustworthy charging scenarios

Cristina Alcaraz *, Javier Lopez, Alberto Garcia

Department of Computer Science, University of Malaga, Campus de Teatinos s/n, Malaga, 29071, Spain

ARTICLE INFO

Keywords:
Multi-agent systems
Situational awareness
Charging stations
Electric vehicles

ABSTRACT

Growing acceptance of Electric Vehicles (EVs) by society is reshaping the transportation sector, which requires the development of robust charging infrastructures. Unfortunately, at present these infrastructures face multiple security challenges due to the complexities involved in integrating information systems with operational systems, which also leads to numerous security vulnerabilities and expands the attack surface. In response to this issue, this paper proposes a distributed Multi-Agent System (MAS) leveraging (i) smart consensus algorithms backed by Opinion Dynamics and (ii) blockchain technology to ensure trustworthy tracking of anomalies in EV charging networks based on the well-known Open Charge Point Protocol. The combined use of techniques and technologies streamlines diagnostic processes at charging stations, but also intensifies the vision (both local and global) necessary for greater protection. The monitoring method goes beyond the usual approaches, which typically focus on network traffic with a local context approach. It is capable of examining local health status related to anomalies found in individual devices, (i) analyzing each operational component and communication link, and (ii) contrasting actual perceptions with those perceived by the neighborhood. This way of broadening the security vision inherently contributes to situational awareness, explaining: where, what, or which components, devices, or charging zones are actually affected. This level of detail can even help ensure a more effective, efficient, and rapid responses depending on the situation.

1. Introduction

The Electric Vehicle (EV) market is consolidating in the global transportation sector. According to the 2025 Global EV Outlook [1], sales of EVs reached 17 million worldwide in 2024, with 3.5 million vehicles sold in 2024 compared to 2023, surpassing total EV sales for the entire year of 2020. Public Charging Stations (CSs) have also doubled in value since 2022, reaching over 5 million in 2024. This demand could even be greater, given that this market is expected to reach $76.31 billion by 2032 [2]. This may be due, in part, to the numerous financial aids provided by governments for clean mobility, such as tax credits, additional incentives, discounts, and exemptions, but also the continuous decrease in the cost of batteries due to the growth in market supply and demand. A key element to maintaining this positive trend is to deploy resilient and secure charging infrastructures, both public and private, which involves numerous stakeholders: CS manufacturers, infrastructure operators, energy companies, the scientific community, electric vehicle owners themselves, and, of course, governments. For example, there is a special support from the EU [3], whose CO_2 emissions standards play a significant role in promoting EV sales.

One of the most commonly used protocols in the deployment of these types of EV charging infrastructures is precisely the OCPP (Open Charge Point Protocol) [4]. It enables the management of a CS network from a centralized system with the capacity to govern and monitor all energy transactions with users. To this end, the central system provides a set of OCPP-assisted remote services, such as user authentication and authorization, power reservations, energy consumption (metering) monitoring, diagnostics, and dynamic configuration of CSs [5]. OCPP is an open source protocol, which has become a de-facto standard supported by the industry in recent years to drive the current EV market. But despite these considerations and the interests of the industry, the numerous security risks that the high demand for electric vehicles and charging infrastructure may bring to current and future charging infrastructure ecosystems [5–9] (including the power grid) must be addressed.

Since charging infrastructures are considered critical in nature [10], CS networks are a very attractive target for attackers, especially because of the economic benefits they may bring to society and the possibility of causing a great impact on its economy. Another determining factor is the deployment of these infrastructures at easily accessible public locations, which exposes CSs to a greater number of threats, such as physical attacks or power theft [5,8]. The type of communication may also encourage attackers to lead potential threats. Some real cases include, for instance, the recent BrokenWire [11], blocking wireless control signals between the EV and the CS to interrupt charging sessions

* Corresponding author.
E-mail addresses: alcaraz@uma.es (C. Alcaraz), javierlopez@uma.es (J. Lopez), alberto_garcia_96@uma.es (A. Garcia).

https://doi.org/10.1016/j.ijcip.2026.100846
Received 10 November 2024; Received in revised form 18 January 2026; Accepted 6 March 2026
Available online 11 March 2026

and cause a Denial of Service (DoS); the botnet specified in [12] consisting of compromised EVs and CSs to overload the power grid; or the side-channel attack in [13] for extraction of private charging session data to lead privacy issues.

Consequently, it has become imperative to offer advanced solutions that provide a better view of what is happening within a large and complex system and control possible cascading effects (*e.g.*, in emergency services assistance). For this reason, the **main contribution** of this article is to present a detection approach based on continuous diagnosis that helps to enhance the properties of Situational Awareness (SA) in large and complex systems, as recommended in [14,15]. The approach considers the relevance of Multi-Agent Systems (MAS) for implementing intelligent OCPP-based charging monitoring solutions, enabling the diagnosis, detection, and monitoring of abnormal events, either by charging device (i.e., in a CS) or by zone or cluster (i.e., more than one CS). For real-time traceability of anomalies and by location area, we also consider the correlation capabilities of Opinion Dynamics (OD) [16], whose values represent health statuses — referred to here as *opinion*. These opinions are generated by software (SW) agents embedded in each CS and enabled with Artificial Intelligence (AI) models for detection such as Machine Learning (ML). For the sake of simplicity, and given that all these technologies must cooperate with each other to lead threat detection and traceability, hereinafter the proposed approach is referred to as MAS-CS (from MAS-based CSs).

Some related work has already demonstrated the feasibility of MAS in similar scenarios to address a variety of issues, and not necessarily related to cybersecurity. In [17,18], SW agents are applied to study user behavior in charging infrastructures in order to derive behavioral patterns in the use of resources. Complex mathematical models have to be used to properly reflect the heterogeneous nature of charging networks and take into consideration all the stakeholders. However, in both papers, agent models are used in the design phase of the charging network to determine the optimal locations of CSs in specific simulated scenarios. In [19], the authors design a MAS to operate alongside the charging network and perform management tasks. It focuses on problems arising from the normal operation management of energy transactions. Likewise, a MAS system, integrated in the management SW of a CS network, is also presented in [20]. Its objective is to automate the negotiations and billing of energy transactions between energy suppliers and users of the charging network. A similar goal is pursued in [21], where an optimization algorithm, implemented by agents, tries to match supply and demand, either by modifying energy prices or by scheduling the charging of some vehicles at less demanded hours in order to avoid exceeding voltage limits. The MAS of [22] shares the same objective, but also takes into consideration use cases of vehicle-to-grid, in which EVs have the ability to contribute energy to the power grid and act as small distributed energy sources. Lastly, in [23] the authors designed a cooperative hierarchical multi-agent system in order to optimize the charging strategy of multiple EV charging stations without prior information of EV arrivals. That system can obtain the optimal solutions for control signals after a single round of communication.

In contrast, we found other authors who studied MAS as a solution for security concerns of distributed power systems, and especially of EV charging networks. For example, paper [24] proposes three different mitigation techniques to defend an EV charging infrastructure against Advanced Persistent Threats (APTs) and compares them in simulated experiments. The best one is a multi-agent deep reinforcement learning comprising agents capable of learning the control policy of each CS controller and restoring the regular operation of infected nodes. Furthermore, the work in [25] introduces a novel attack called "for-purpose attack", where the attackers inject fake data to deceive the system and gain an economic benefit. The target under attack is a distributed and decentralized multi-agent optimization algorithm that controls an EV charging infrastructure. The work contains theoretical and mathematical demonstrations about the feasibility of these attacks. In [26], authors present two control schemes for distributed energy storage systems under cyber attacks. These schemes are agent-based and follow a consensus algorithm to detect information deviations. Both schemes ensure optimal control operations in an unreliable communication network.

Similarly, the paper [27] analyzes the generic benefits of applying MAS to bring cybersecurity to Operational Technology (OT) and IoT networks. The authors also propose a specific architecture for MAS in OT environments that suggests the agents' components, their communication language and their different interactions. A multi-agent approach is used in [28] to model the attacker-defender game in critical infrastructure scenarios. The solution goes through the use of reinforcement learning in simulations for defenders to learn the best configurations and prevention actions in order to address the system's vulnerabilities. Another multi-agent-based hierarchical detection and mitigation scheme for power systems to defend against physical faults and cyber attacks is depicted in [29]. The detection is based on a rule-based engine and the MAS consists of three layers of agents where lower layers manage physical attributes and upper layers work with redundancy network information. Lastly, [30] aims at detecting specific fake data injection attacks to distance relays, a critical component in Smart Grids. For this purpose, a multi-Agent distributed deep learning algorithm is developed to detect if an attacker has injected false fake data to pretend a fault or if it is really a system fault.

Unfortunately, none of the previous works (also summarized in Table 1) propose a MAS that addresses the importance of tracking potential attacks for the protection of EV charging infrastructures, considering: (i) the operational performance of each component integrated into each CS together with its (ii) communication links, in addition to the (iii) surrounding status of each CS. The closest related works are [29,32]. The study [29] generalizes the application of MAS protection systems to power systems and does not show their relationship with specific standards and protocols for EV charging such as OCPP; whereas [32] rely on centralized architectures (such as cloud computing) to calculate the contextual state of each area. This centralization (also remarked in [31,33]) is due to the type of intelligence implemented, which applies the most innovative forms of AI, such as large language models or reinforcement learning. But if this capability were spread out among CSs with limited computational resources, as proposed in this paper, it would significantly penalize real-time monitoring by device, area, and overall. For that reason, MAS-CS aims to deploy simple software (SW) agents on each CS to dynamically inspect anomalous events related to the health status of each observed device. This involves using a (lightweight) unsupervised ML model (for autonomy and lightness, as also indicated in [34] for critical scenarios) and sharing-based consensus strategies for real-time monitoring without exploring other related security events to reduce the scope of the investigation — which is also sufficient to verify that the approach is useful for a key subset of events. Specifically, all *individual opinions* are shared with neighboring agents to calculate the level of anomaly per area (*neighborhood opinion*) and correlated in a central system to later calculate the overall status of the entire charging infrastructure (*global opinion*). Thus, these three levels of opinion allow to have a holistic view of the system, contributing not only to SA, but also to the real-time protection of critical assets.

The paper is structured as follows. Section 2 features the MAS-CS architecture together with the deployment of its agents, enabled with an ML model to estimate the individual opinions of their respective CSs. As detailed in Sections 3.2 and 3.1.2, this local opinion is later shared with the surrounding agents to calculate the neighborhood opinion (per zone), whose value is essential for tracking of threats. In order to demonstrate the usefulness of diagnostics for governance and control actions, and the possibility of extending the approach in the future, a new agent role in charge of monitoring abuses in the charging network, such as energy theft or DoS, is defined in Section 3.3. Finally, practical demonstrations are carried out in Section 4 through a virtual testbed called *Urban Lab*, whereas Section 5 concludes the paper and outlines future work.

Table 1
Related work and main differences between approaches.

Ref.	Year	Target system	Main purpose	Security oriented	Diagnosis[a]	Detection[b]	Tracking[c]
[17]	2019	EV Charging Infr.	Discover user behavior patterns	No	Partial	Global	Faults
[18]	2017	EV Charging Infr.	Discover user behavior patterns	No	Partial	Area	Faults
[19]	2012	EV Charging Infr.	Operations management	No	Partial	Global	Faults
[20]	2015	EV Charging Infr.	Automation of negotiation and billing	No	Partial	Local	Faults
[21]	2014	EV Charging Infr.	Match energy supply–demand	No	Partial	Global	Faults
[22]	2011	EV Charging Infr.	Match energy supply–demand	No	Partial	Global	Faults
[23]	2022	EV Charging Infr.	Match energy supply–demand	No	Partial	Global	Faults
[24]	2022	EV Charging Infr.	Mitigation against disruptions caused by APTs	Yes	Partial	Local	Attacks
[25]	2023	EV Charging Infr.	Novel attack study	Yes	Partial	Global	Attacks
[26]	2017	Distributed Energy Storage Systems	Detection and mitigation of attacks	Yes	Partial	Local	Attacks
[27]	2018	OT/IoT Networks	Design of a MAS architecture for OT/IoT security	Yes	Partial	Area	Attacks
[28]	2018	Critical Infrastructures	Address vulnerabilities in Critical Infrastructures	Yes	Partial	Area	Attacks
[29]	2020	Power Systems	Detection and mitigation of attacks and faults	Yes	Complete	Area	Both
[30]	2023	Smart Grids	Detection of faults and attacks	Yes	Partial	Area	Attacks
[31]	2024	Power System & EV Charging Infr.	Centralized monitoring in power grid and response to Charging Inf.	Yes	Partial	Local	Attack
[32]	2025	EV Charging Inf.	Centralized detection in the CSMS	Yes	Complete	Global	Both
[33]	2025	EV Charging Inf.	Collaborative detection in a hierarchical computing infr.	Yes	Partial	Global	Attacks
Ours	2026	EV Charging Infr.	Distributed detection in the entire ecosystem	Yes	Complete	Global	Both

[a] Diagnosis: **partial** (analyzes one type of system event: operational, resource utilization or comm. channels), **complete** (exhaustive analysis of the system and its surrounding).
[b] Detection: **local** (does correlate events or statuses of individual nodes), **area** (correlates events in nodes grouped by zones), **global** (aggregates the events across the entire net.).
[c] Tracking: **faults** (looks for unintentional system faults), **attacks** (looks for deliberate security events occasioned by malicious actors), **both** (casual faults and attacks).

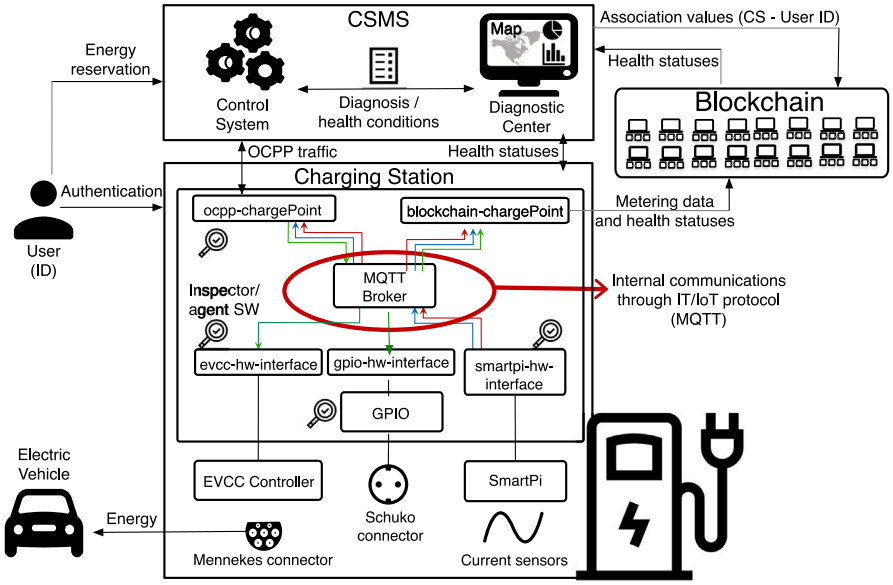

Fig. 1. MAS-CS architecture and interactions between components.

2. MAS-CS architecture and testbed

In this section, we propose the MAS-CS architecture based on the deployment of a set of SW agents capable of dynamically performing diagnosis processes in each CS, detecting and tracking anomalies to bring the concept of SA closer to the charging environments. The design, illustrated in Fig. 1, is also based on the standardized OCPP specification detailed in [35]. The specification establishes that each CS is capable of supplying energy to EVs if users are properly authenticated, either through traditional authentication mechanisms (use of IDs, tokens or PINs), mobile services or through pre-reserves for a given EVSE (Electric Vehicle Supply Equipment). All these authentication procedures are managed by the Charging Station Management System (CSMS) to authorize the energy charging sessions. In our approach, the CSMS is also capable of controlling all telemetry tasks and monitoring CS health statuses.

Unfortunately, the OCPP diagnostic procedures normally follow systematic actions based on C&C (Command and Control) without contemplating automatic procedures that dynamically determine health statuses of remote CSs. To address this issue, our approach incorporates a *diagnostic center* located within the CSMS itself to centralize and dynamically correlate health statuses. This center, also depicted in Fig. 1, is capable of (i) periodically collecting activity events through SW agents deployed in each CS, such as status of the controller, network and EVSE connectors, time and frequency of consumption, etc.; and (ii) alerting and visualizing critical situations (*e.g.*, through statistical figures and geolocation maps), thereby favoring decision making. To ensure the immutability of historical data, the approach considers the incorporation of a permissioned blockchain network as recommended in [36]. Through this network, it is possible to ensure in the future that related energy utilities and providers can interact in a common environment to not only improve the tasks of control and distribution of energy, but also to fairly assist in billing tasks by identifying possible fraud or energy theft (*e.g.*, disproportionate energy consumption by a specific user ID).

More specifically, each CS must be able to collect charging-related events to determine the actual health level within its own system. These events are provided by electrical sensors including the smart meter, to be forwarded later to the blockchain in the form of periodic reports. When an energy transaction is completed, the CSMS also transfers the

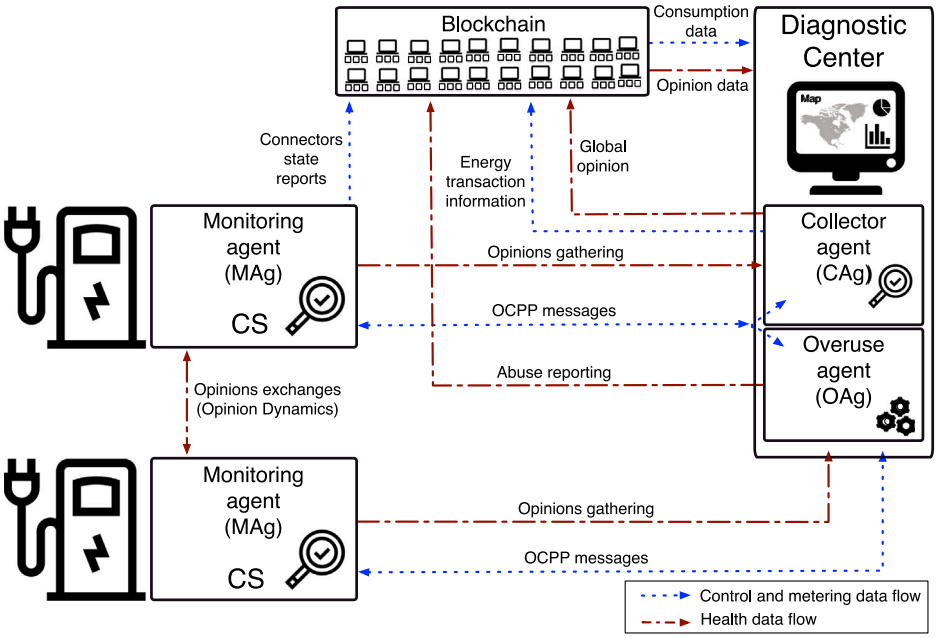

Fig. 2. Integration of SW agents in MAS-CS.

statistical values related to that transaction to the blockchain, associating the record with the CS and the user ID involved in the transaction. This information is displayed on a regular basis in an interactive panel to explain in more detail what is happening within one or several CSs. Moreover, Urban Lab, the virtual testbed implemented for MAS-CS, assigns a set of independent SW modules to each CS to comply with the modularity principles. These modules, interacting with each other under the inclusion of the Message Queue Telemetry Transport (MQTT) protocol, have the following functions:

- *ocpp-chargePoint*: manages OCPP transactions with the CSMS, keeping a record of active energy transactions taking place on the CS.
- *smartpi-hw-interface*: collects power consumption values from sensors installed in the CS and transfers them to the rest of the modules for further processing.
- *blockchain-chargePoint*: forwards to the blockchain the data sent by the other SW modules via the MQTT broker, such as: metering values, performance information or control data.
- *evcc-hw-interface*: controls the activation/deactivation of a medium-rate Mennekes connector, present in the CS, through the EVCC (Electric Vehicle Charger Controller).
- *gpio-hw-interface*: controls the activation/deactivation of the slow-rate Schuko connector, present in the CS, through the GPIO (General Purpose Input/Output) module.

To show the role of the MAS within MAS-CS, Fig. 2 envisions the deployment of a set of SW agents. Each agent may have the ability to interact and exchange information with other agents and coordinate its actions for the diagnosis. Depending on the type of action, we can find three types of SW agents: (i) the *Monitoring Agent* (MAg), (ii) the *Collect Agent* (CAg), and (iii) the *Overuse Agent* (OAg). The former is hosted in each CS to obtain local measurements related to computational and communication performance, in order to subsequently estimate its health status (opinion). In contrast, the CAg is located in the CSMS itself to collect feedback from the MAgs deployed in each CS. Once the opinions are gathered, the CAg processes them to calculate global and local health indicators, either on a particular CS, a group of CSs or the entire charging network. The OAg is also in the CSMS, and operates independently to derive possible abuses carried out by system users with respect to actual energy consumption. To do so, it takes advantage of the information received and processed by the previous agents to deduce possible abuses or fraud. The communication of these three agents is also illustrated in Fig. 2, where two types of data are distinguished: (i) those related to OCPP transactions for control and authorization for charging, and (ii) those related to opinion values for diagnosis. Both flows are independent from each other and use different communication channels and protocols to avoid mixing operational and diagnostic operations.

3. Main functions of the agents in MAS-CS

As indicated in the previous section, we consider three relevant agents integrated in the MAS-CS system, which contribute to SA by managing and tracing anomalous events.

3.1. Monitoring agent: individual opinion in MAS-CS

As previously mentioned, a MAg is a SW agent capable of monitoring the use of the CS's HW resources, and of checking the status of its connections with other CSs that are physically close (neighbor stations). In Urban Lab, this agent takes measurements of some parameters, such as CPU usage, the amount of storage used, the Round-Trip Time (RTT) of packets to the neighbor stations, and the percentage of packet losses. With these measurements, each agent executes an ML model to predict the local opinion, indicating the probability that the CS is compromised or out of its normal operational threshold. To lead consensus about health status within a neighborhood, the OD technique [16] is considered. The technique is able to dynamically calculate neighborhood opinion by merging individual CS opinions, improving diagnostic tasks and activity tracking.

As all these functional characteristics are attractive to have a better understanding of the situation, the following details the relevance of anomaly detection for local opinion, and the OD for neighborhood opinion by charging zone.

3.1.1. Individual opinion through anomaly detection

Anomaly detection is a simple way to contribute to SA, tracking potential threats [16] both within and between CSs, and showing at all times which charging zones are being affected. This requires

incorporating ML models in each MAg integrated in the MAS-CS, whose input variables contain health statuses (normal or abnormal) collected periodically by these agents. These variables, processed as *health features* of the HW and SW components of a CS, are calculated as follows:

- *Processor usage* (ρ): it is associated to the computational capacity present in each CS, it gives an indication of the level of overload in each station. Due to the normal operations of the CS and its main communication interfaces with the CSMS, the processor workload should remain relatively constant. For its computation, Eq. (1) considers the period of time that the processor has been occupied (t_{busy}) by a SW process from the last measurement made by the MAg, also including periods of inactivity (t_{idle}), such that:

$$\rho = \frac{t_{busy}}{t_{busy} + t_{idle}} 100 \tag{1}$$

- *Memory usage* (μ): RAM usage is another key element for the execution of any program or system process. Its value must be monitored when trying to discover any casual or intentional actions that corrupt the navigation or use of critical sections of memory. Eq. (2) calculates this value by measuring the percentage between the amount of memory in use ($bytes_{available}$) and the total memory capacity ($bytes_{total}$), such that:

$$\mu = \frac{bytes_{total} - bytes_{available}}{bytes_{total}} 100 \tag{2}$$

- *Storage usage* (δ): Eq. (3) focuses on measuring the amount of storage available on the hard disk, considering the actual space used ($bytes_{used}$) with respect to the total amount of memory in the resource ($bytes_{total}$). Some DoS attacks aim to delete or make multiple copies of data on a disk so as to exhaust storage capacity. Thus, constant monitoring of the δ value helps human operators control and manage the level of overflow when necessary.

$$\delta = \frac{bytes_{used}}{bytes_{total}} 100 \tag{3}$$

- *Round-trip time* ($\overline{RTT}$): regarding the status of the TCP/IP connection with neighboring CSs. Specifically, this feature refers to the duration (measured in milliseconds in Urban Lab) from the time a request is sent to a CS until the response is received from the CS. This value can be applied by anomaly detectors embedded in MAg to: (i) predict anomalous events occurring in the CS network, (ii) determine the level of latency of the connection, and (iii) measure the actual performance of an online diagnostic service [37]. A use case may be the detection of MitM attacks or rogue devices stealthily installed in the charging networks. This action would result in a significant increase of the $\overline{RTT}$ feature between two legitimate nodes of the network, whose value is computed according to Eq. (4). This equation averages the RTT_i measurements observed after a sequence of N request packets between two CSs.

$$\overline{RTT} = \frac{\sum_i^N RTT_i}{N} \tag{4}$$

- *Time-To-Live* ($\overline{TTL}$): this is the typical IP packet lifetime estimation parameter, the value of which is included in the packet header to control the undefined flow of network traffic [38]. Eq. (5) computes this value to detect changes in the network topology, considering the number of hops a packet can make before it is discarded by a router (corresponding to TTL_i) and the number of packets N sent in each measurement process.

$$\overline{TTL} = \frac{\sum_i^N TTL_i}{N} \tag{5}$$

- *Packets loss* ($PacketLoss$): as indicated in Eq. (6), this variable is measured as the number of requests ($n_{request}$) that have not received a response ($n_{request} - n_{reply}$) over the total number of requests sent. Moreover, its value can be easily calculated by considering the typical ping for the heartbeat, in order to prevent DoS attacks on-the-path, such as blackhole, greyhole, sybil or wormhole attacks. In the latter two cases, malicious nodes may strategically advertise themselves as the best route to a certain destination, intercepting and discarding packets at their convenience [39].

$$PacketLoss = \frac{n_{request} - n_{reply}}{n_{request}} 100 \tag{6}$$

While these are the main health features that the MAS-CS applies as an input to the anomaly detector, others may be equally effective to provide further evidence in the future. Some of these features could be, for example, the volume of traffic, the type of communication protocols observed in a network interface or the port used, and the alerts generated by an intrusion detection system. For the dynamic detection in each MAg, we have adapted the Local Outlier Factor (LOF) model, as it is considered an efficient ML algorithm for critical scenarios [34,40,41] where speed of response and simplicity in the detection process are required [42]. LOF is an unsupervised learning algorithm focused on calculating an anomaly score for each sample by measuring its local deviation from the nearest samples and using the k-nearest algorithm. When a sample has a lower local density than the others, it is considered an outlier [43].

As LOF is an unsupervised detector, it must be able to learn the different patterns of behavior in real time without having labeled previous samples. That is, it must be able to differentiate valid operations from behavior that deviates from the learned behavior. To do this, the detector needs an initial training phase in which it does not make predictions, but simply obtains and memorizes measurements that are considered the normal behavior of the CS. Although this pre-learning phase has been implemented in MAS-CS in such a way that it can be parameterized depending on the number of training observations to be collected and the interval to be set between two consecutive samples, having a large number of training observations in order to guarantee accurate predictions is recommended [43].

After this pre-learning phase, the LOF-based detector is trained with the training observations. Hereafter, all measurements captured by the MAg are evaluated and receive an anomaly score corresponding to the individual opinion, the value of which ranges from zero to one (*i.e.*, [0-1]). Values close to one indicate that the CS is operating normally, while values close to zero indicate that the monitoring agent is detecting an anomalous status that deviates from normal a operation situation. To update the opinion of the CS_i at a given time t, $\Delta_{opinion}$ is computed to determine the level of deviation of the new opinion value ($x_i[t+1]$) from the opinion prior to the anomaly detector prediction ($x_i[t]$). Thus, the new opinion value is obtained by simply adding this $\Delta_{opinion}$ to the current opinion, i.e:

$$x_i[t+1] = x_i[t] + \Delta_{opinion} \tag{7}$$

In addition to the anomaly detector, the MAg also has a mechanism to check the integrity of data in CSs (*e.g.*, service scripts, configuration files, etc.). This allows the SW agent to further penalize its opinion value if malicious actions are detected.

3.1.2. Neighborhood opinion through OD

The group opinion is computed thanks to the consensus managed by charging zone. The idea itself stems from how to manage individual opinions within a given society, where each opinion can be influenced by the opinion of other individuals (SW agents) living in the same society. Evidently, any society can end up fragmented into different local opinions, so that agents who present a similar view tend to establish consensus and converge to the same opinion [16]. As noted in previous sections, this information benefits monitoring systems by allowing them to anticipate situations and make more accurate decisions to respond

accordingly. This also means that the more SW agents are deployed in CS networks and reporting similar opinions, the more trustworthy the diagnostic value will be. To achieve this level of accuracy, it is also necessary to consider the assumption that CSs should be deployed in close proximity to each other, so their agents may be able to provide similar opinions when experiencing similar situations or threats [16].

On the other hand, the OD model integrated in MAS-CS run periodically, even more frequently than the predictions of the anomaly detector (*i.e.*, the OD invocation takes place between two consecutive predictions). In order to calculate the OD, each SW agent must know the opinion of the neighboring CS agents. For this purpose, a request–response mechanism is integrated so that each agent can request the opinion of its neighbors. Each SW agent will act as a client when requesting the opinion of a neighboring agent, and as a server when requested to provide an opinion by its own neighbors. When an agent does not receive a reply with its neighbor's opinion, it will consider that it is in the worst case scenario and will assume that its neighbor's opinion is zero. This is intended to detect cases where the attacker tries to stop the execution of a SW agent in order to subsequently avoid being detected by the MAS-CS system.

So far, there are several ways to manage consensus [44], but for the sake of simplicity and a rapid diagnosis, we consider the technique defined in [16] for critical scenarios. Basically, the approach aims to calculate weighted average following a uniformity criterion. Each SW agent gives weights uniformly to other agents of its neighborhood when they have a similar opinion to its own. Likewise, two agents are considered to have a similar opinion when the difference between their values is less than a threshold ε [16]. In other words, if we have a SW agent, i, whose opinions are stored in $x_i[t]$, its new opinion after an iteration of ODs, $x_i[t+1]$, will also depend on the opinions, $x_j[t]$, of those agents $j \in J$ within its neighborhood (J being the cardinal of the CS network) and complying with the condition: $\mid x_i[t] - x_j[t] \mid \leq \varepsilon$. It is worth mentioning that the agent i is also part of J since its own opinion also interferes in the group opinion as illustrated in Eq. (8):

$$\begin{aligned} x_i[t+1] &= x_0[t]w_{i0} + x_1[t]w_{i1} + \cdots + x_{N-1}[t]w_{i(N-1)} \\ &= \sum_j x_j[t]w_{ij} \end{aligned} \tag{8}$$

here, w_{ij} is the weight that agent, i, gives to its neighbor, j, such that:

$$w_{ij} = \frac{1}{\mid J \mid} = \frac{1}{N} \tag{9}$$

Algorithm 1 reflects the actions mentioned in the previous sections, where each MAg captures a set of functional parameters and calculates the local opinion by means of an anomaly detector that must be previously trained to provide correct predictions. Subsequently, each opinion is shared with the rest of the SW agents integrated in the neighboring CSs; and if no response is received, the worst case scenario is assumed.

In order to demonstrate the viability of Algorithm 1 and its generalization to various broader contexts, the corresponding correctness proof is presented. This proof comprises two relevant verification conditions: (i) validity, which guarantees local opinion when required by a CS, and (ii) termination, which verifies that the algorithm terminates in a finite time, regardless of the number of functional characteristics observed within a CS and with respect to its operational environment. The precondition assumes that the system is based on at least one or more CSs under a finite number of CSs to be realistic, all equipped with an Anomaly Detector (AD) to internally calculate the opinion on the health status; while the postcondition considers the capacity of the system for calculating and transferring (if applicable) the opinion according to the neighborhood. Taking both conditions into account and by induction, we demonstrate both validity and completeness:

Algorithm 1 The process flow of monitoring agents

```
x: CS opinion vector; host: local CS;
neighbors: remote CS vector; AD: Anomaly Detector;

while true do
    rParams ← measureResources
    for neighbor in neighbors do
        cParams ← measureComm(neighbor)
    save (rParams, cParams)
    if idlePhase is true then
        iteration ← iteration + 1
        if iteration > idleIterations then
            train AD(rParams, cParams); idlePhase ← false
    else if idlePhase is false then
        anomalyScore ← AD(rParams, cParams)
        x(host) ← updateOpinion(anomalyScore); save x(host)
    for opinionDynamicsIterations do
        for neighbor in neighbors do
            x(neighbor) ← requestOpinion(neighbor)
            if no response then x(neighbor) ← 0
        x(hosts) ← computeOpinionDynamics(x)
        save x(host); wait opinionDynamicsInterval
    wait anomalyDetectionInterval
```

Case 1: Let CS be the set of charging stations whose cardinality is reduced to a single unit, such that $|CS| = 1$ and $|neighbors| = 0$, since the set of CSs deployed near a location (or that are part of an operational cluster) is consequently null. In these circumstances, Algorithm 1 calculates only the opinion of the CS without transferring its value to other related devices, since $|neighbors| = 0$.

Induction: $|CS| > 1$ such that each CS $\in$ *neighbors* and $|neighbors| \neq 0$ in a zone or cluster. In this case, Algorithm 1 calculates the local opinion of each observed CS. For each CS in the set of *neighbors*, the reach status is calculated, as well as the deviation level of each of the parameters considered relevant for local monitoring and diagnosis — *cf.* Section 3.1.1. However, to provide a true value for the status of a node, the opinion of each CS must be correlated with the opinions of neighboring nodes, thereby seeking maximum accuracy in the diagnosis. As the set is limited to a finite number of CSs, in each iteration of the loop the set *neighbors* is updated to: $neighbors \leftarrow neighbors \setminus CS$, until $neighbor = \emptyset$.

As can be seen, Algorithm 1 terminates and guarantees the calculation of the local opinion for each CS. If we additionally examine the complexity of the approach, it reaches $O(n^3)$ in the worst case. However, the core of the algorithm and the actual calculation of opinions actually reach a complexity of $O(n^2)$, since the first loop represents the way to keep the MAS-CS life cycle alive. Obviously, to maintain the simplicity of the algorithm and achieve real-time detection (by agent) with minimal or no human intervention, the AD must also be simple as LOF. Nonetheless, any other unsupervised algorithm with high accuracy and simplicity can be equally valid.

3.2. Collector agent: a global opinion

In the MAS-CS, the CAg corresponds to a SW process centralized in the CSMS, capable of collecting all the opinions collected by the MAgs. To do so, it follows similar mechanisms to the exchange of neighborhood opinions, but this time the CAg (acting as client) acts individually and periodically requests a local opinion from each MAg. In the event that the CAg requests a local opinion from an MAg and gets

no response from the latter, this is considered the worst case scenario. This means that the MAg's opinion is assigned a value of zero, since in a normal operating situation, all MAgs should be available to respond to the requests from the CAg.

After obtaining feedback from all MAgs in each iteration, CAg calculates the average of all health statuses received from the entire charging infrastructure. This calculation may vary depending on the type of approach adopted: (i) either by averaging all local opinions across the entire CS infrastructure; or (ii) by averaging by charging zone. The latter case makes sense when charging infrastructures are large and it is necessary to track the progress of a threat between zones, thus ensuring decision making. These decisions can even be based on historical data, whereby all Urban Lab data is sent to the Hyperledger Besu-based blockchain for further analysis, auditing and accountability [36], and composed of five validator nodes. Algorithm 2 shows the performance of CAg and its potential features following the second approach described above.

Algorithm 2 The process flow of the collector agent

```
x: CS opinion vector; y: zone opinion vector; z: global opinion;
clusters: subsets of CSs that represent zones
while true do
    for cluster in clusters do
        for CS in cluster do
            x(CS) ← requestOpinion(CS)
            if no response then
                x(CS) ← 0
            save x(CS)
        y(cluster) ← computeZoneOpinon(x)
        save y(cluster)
    z ← computeGlobalOpinion(y)
    save z; wait pollingInterval
```

Similar to the analysis performed for Algorithm 1, the following analysis details the proof of correctness that demonstrates the viability and generalization of Algorithm 2 for its application in broader contexts. In this case, the demonstration proves two relevant conditions: (i) validity, guaranteeing the calculation of the opinion both in global terms and by zone; and (ii) termination, verifying that the algorithm terminates in a finite time regardless of the number of CS and clusters. As a precondition, we establish that the system is based on at least one cluster composed of one or more CSs per zone, all equipped with an AD as detailed in Algorithm 1; while the postcondition takes into account the capacity of the system by calculating (through OD) the overall opinion per zone — until all clusters in the charging infrastructure are processed, *i.e.*, until the set $clusters = \emptyset$. By induction, then:

Case 1: $|CS| = 1$. This means that the system has a single CS located in a deployment zone, such that $|cluster| = 1$, $|neighbors| = \emptyset$, and $|clusters| = 1$. When Algorithm 2 is initiated, CAg requests the local opinion (*cf.*, Algorithm 1 - Case 1) of the CS deployed in $cluster$. In the new iteration, the set $cluster$ is updated ($cluster \leftarrow cluster \setminus CS$) reaching its value $\emptyset$, but also the set $clusters \leftarrow clusters \setminus cluster$. After executing both iterations (corresponding to *for*), the CAg calculates the global opinion, which depends on the only opinion received by the CS. If CAg does not receive the corresponding value, it automatically considers it null to leave evidence of a serious occurrence in the area.

Case 2: $|CS| \geq 1$, $|cluster| \geq 1$, and $|clusters| = 1$. The algorithm performs in the same way as described in Case 1, except that the third iteration depends on the number of CSs in $cluster$. In this process, individual opinions (as indicated in Case 1) are collected to calculate the corresponding average, and in each calculation, the set is updated until it reaches the empty value. In addition, as $|clusters| = 1$, the iteration forces $clusters \leftarrow clusters \setminus cluster$, reaching its empty value and the completion of the second loop.

Induction: It is assumed that several CS ($|CS| > 1$) are deployed in different charging zones, such that $|clusters| \geq 1$. This means that, while $clusters \neq 0$ (second iteration), the CAg processes the local opinion of each CS included in each cluster, which also $\in clusters$, and updates the opinion extracted per zone — as detailed in Case 2. In each iteration, the set of $clusters$ is updated until it reaches its empty value. Depending on the number of clusters to be examined, the third iteration will run as many times as necessary until the set itself reaches the empty value. Given that the precondition establishes that the number of CSs is finite in order to illustrate more realistic scenarios, the completion of both loops (in relation to the *for*) is validated. After both loops, the calculation of the global opinion is computed and saved.

Therefore, Algorithm 2 also terminates and guarantees the calculation of the opinion in its different SA views: by device (local) and zone (cluster). If we examine the algorithm, we can also note that its complexity is $O(n^3)$. However, the core of the algorithm and the actual calculation of opinions is based on $O(n^2)$, since the first loop represents the way to maintain the vitality and autonomy of CAg — equivalent to Algorithm 1.

3.3. Overuse agent: Facing charging resources abuse

By leveraging the MAS-CS data and the role of the OAg, it is now possible to extend security functionalities. In charging networks, both legitimate end-users and human operators may misuse charging resources to steal energy or cause some kind of unavailability, perhaps to monopolize the CS network's essential services. The latter may result, for example, in stations being kept busy for an excessive amount of time, with the consequence that other users are unable to charge their vehicles. To reduce the impact, OAg monitors all OCPP energy transactions that take place inside and outside each station. For example, in the Urban Lab testbed, this transaction monitoring would be done over the MQTT protocol (see Fig. 2), whose messages contain the duration of the OCPP transactions, the ID of the CS where they take place and the ID of the user associated to the OCPP transaction [5]. With this information, the OAg can calculate the cumulative time during which each system user uses the charging network, ensuring that it does not exceed a certain pre-set usage threshold. When a user exceeds the usage limit and the policies set by the organization, a default penalty should be imposed. An example of a penalty could be a ban on further use of the system for a period of time, proportional to the length of time of abuse. So each time a user requests starting a new energy transaction, the OAg in the CSMS would first check whether the user's ID has already been penalized. If so, the OAg would notify the CSMS to interrupt the process of starting the energy transaction.

Penalty policies can be very diverse. For example, the OAg could take into account those cases where a user unintentionally exceeds a certain time in the use of services. The agent will not penalize user IDs whose charging time has not exceeded a pre-determined courtesy time, up to a pre-defined number of consecutive times. In these situations, the user will simply be notified of the violation and urged not to repeat it. Likewise, the OAg must also control the sanctions and readmit users whose sanctions have expired. Therefore, the OAg aims to create a dissuasive effect, which tries to make users aware of the proper use of the system and prevent the existence of users in the system who repeatedly break the rules.

4. Attack model and experimentation in urban lab

As stated above, the simulations have been carried out on the Urban Lab virtual testbed. The testbed visualizes all the elements, including the SW agents involved in the charging network. For this purpose, a network of three CSs (CS-1, CS-2, and CS-3) has been deployed and

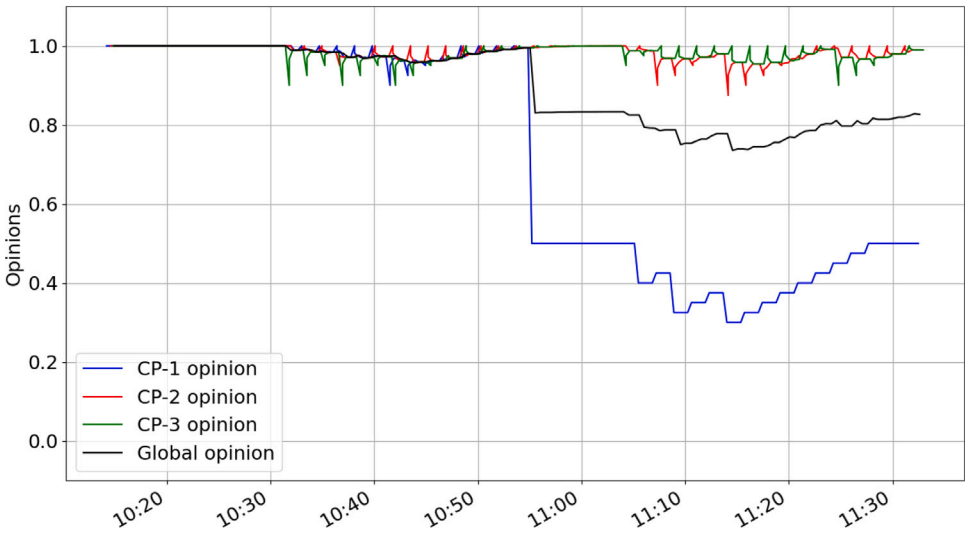

Fig. 3. Opinions of CSs after an attack on integrity (EC-1)

connected to a virtual CSMS. Each node runs all the components that have been defined in previous sections, including those that connect them to the blockchain for data collection. An instance of the MAg is deployed in each of the virtual CSs, while the CAg is configured in the virtual CSMS in charge of calculating the overall opinion, following the first approach defined in Section 3.2. Inspired by the study conducted in [5], two Experimental Cases (ECs) have been designed to produce attacks against the integrity and availability of critical resources, threats considered to be the most influential in charging scenarios [5]. In turn, these two ECs have been in turn designed to assess the degree of detection of MAS-CS, such that:

- *EC-1, an attack on the integrity of CS-1*: the attack focuses on modifying the *blockchain-chargePoint* component (see Section 2 and Fig. 1) of CS-1 with the aim of reporting incorrect metering data to Urban Lab's blockchain. The consequences could clearly reflect an impact on the management and billing of end-users, and subsequently lead to possible fraud or energy theft.
- *EC-2, a DoS in CS-1 with a cascading effect on the availability of other CSs*: the main objective is to cause a DoS on the CS-1 EVSE network. The consequences of its effect go beyond the inconvenience of not providing services to end-users, as the unexpected shutdown of EVSE may cause unforeseen damage to connected EVs or to the CS itself. In addition, in order to have a greater impact on EC-2 and provide a broader discussion, the attack is extended to address a cascading effect against the availability of other CSs, where not only CS-1 is rendered unavailable, but also CS-2.

In both ECs, we have assumed the occurrence of an insider (ID_m) with the ability to access the EVSE network and escalate privileges to carry out its goals. In the case of EC-1, the goal is to manipulate the *blockchain-chargePoint* component to intentionally change the metering data that is sent internally in the CS (using OCPP over MQTT) and in favor of the user ID_m; whereas the goal in EC-2 is to disrupt access to EVSE connectors for the availability of stations. In both circumstances, the threat must be detected with MAg and CAg, and the ID_m must be penalized accordingly through OAg.

Starting with EC-1, Fig. 3 illustrates the MAgs' opinions deployed in CS-[1-3]. We observe that before the threat occurs at CS-1, the opinions of the three stations are very close to one. Once the attack occurs, CS-1's MAg detects the malicious influence causing CS-1's opinion value to drop sharply. This decrease also has an impact on the global opinion, although to a lesser extent, as this value reflects the overall health status of the three CSs as a whole. Similarly, the opinions of CS-2 and CS-3 have not been strongly altered, as the threat has only been focused on CS-1 without being detected by the surrounding MAgs. In addition, due to the neighborhood opinion through OD, when the CS-1's feedback is too abrupt with respect to its neighbors' feedback, society is fragmented into different opinion groups, creating convergences without merging with each other. This, in turn, allows administrators and security personnel to pinpoint the most influential areas and act accordingly in a timely manner.

For EC-2, Fig. 4 shows the opinions of each of the CSs during the execution of the attack. The graphs embedded in the figure show the exact moments when each part of the attack is detected. When the CS-1 ceases to be operational, its MAg also ceases to be operational, and no longer provides opinion values. At that moment, the MAgs of CS-2 and CS-3 detect that they have lost the connection with CS-1 and cease to receive feedback from their MAg. This situation, in turn, seriously affects CS-2's and CS-3's opinions, and an anomaly is detected. This decrease of opinions in CSs, and the impossibility of receiving feedback from CS-1, causes the global opinion in the CSMS to vary accordingly. As a complement to this study, Fig. 5 shows the RTT measurements taken by the MAgs for the anomaly detection. The embedded graphs show that after the first phase of the attack, the RTT value of the packets destined for CS-1 reaches zero. Subsequently, the second phase of the attack, *i.e.*, the cascading DoS, takes place and CS-2 ceases to operate. As a result, CS-3 is the only operational station, reporting not only on the situation but drastically changing the global opinion as well (see Fig. 4).

Thus, Figs. 4 and 5 also serve to show how slight fluctuations sometimes appear in agents' opinions, mainly due to changes in the RTT of the packets transiting the network. Since the CSs are close to each other (they are virtualized on the same server), the RTT is almost exclusively composed of the processing time of the packets at source and destination (and not of the propagation delay in the network), which in turn depends on the workload of both nodes at that particular time. In the RTT measures this is shown as peaks of very short duration, which means that the opinion does not change significantly. In turn, it can also be seen how OD helps to overcome this problem by averaging the slightly different opinions. Last but not least, it is recommended that all application scenarios (including those related to EC-1 and EC-2) apply appropriate corrective measures, such as isolation or penalties. It is possible to regulate bad practices (see Section 3.3) through updated security policies. Abusive use by IDs can be controlled by permanently/temporarily denying their access to CSs and adding such IDs to blacklists, both for billing and control purposes.

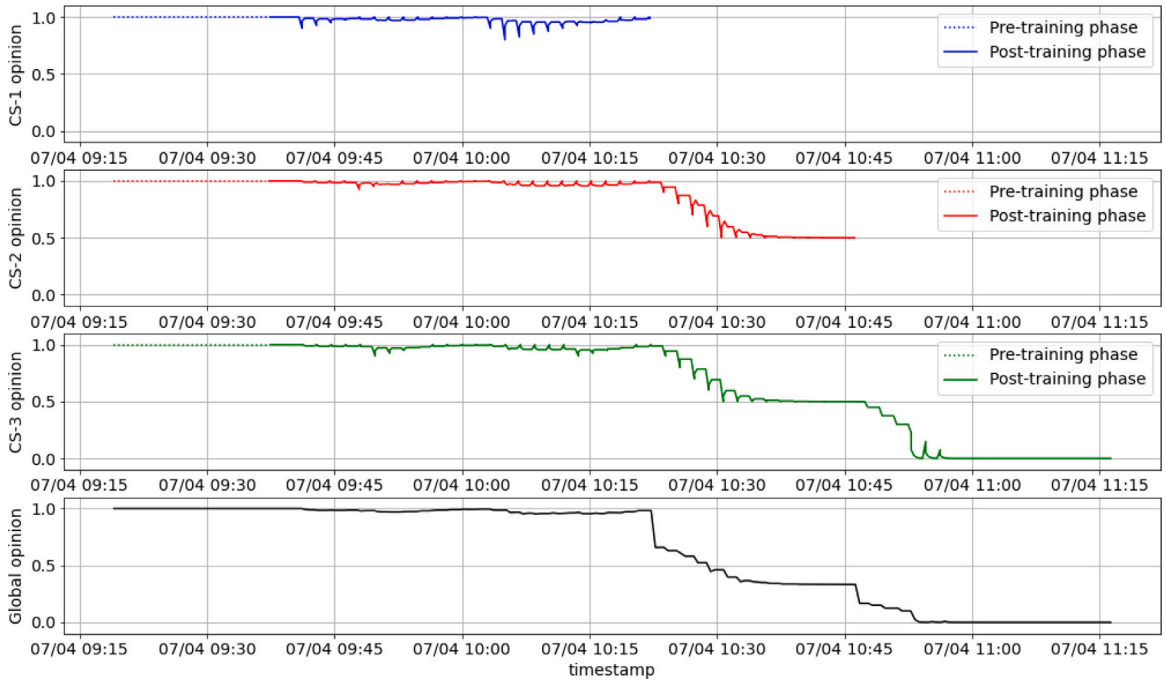

Fig. 4. Opinions of CSs after an attack on availability (EC-2)

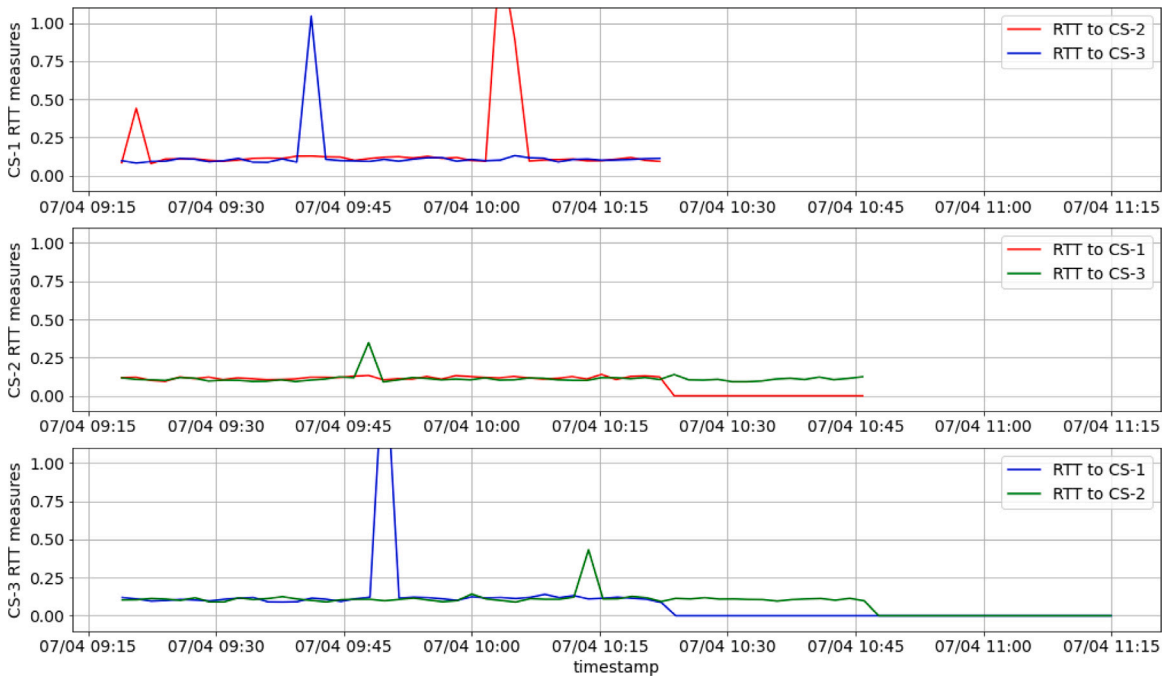

Fig. 5. RTT in presence of an attack on availability (EC-2)

5. Conclusions

In this work, we have demonstrated the usefulness of multi-agent systems for deriving health deviations in CS networks, while increasing the situational awareness of the whole system. To this end, three types of SW agents have been deployed not only to derive deviations through anomaly-based detection and consensus, but also to lead penalties if fraudulent abuses are detected. At the same time, the hierarchical deployment of the CSs and the management of different opinions per area makes it possible to manage and trace different levels of affection in real-time, whether by CS, a group of CSs or at a global level.

The experiments demonstrate the feasibility of the approach for a limited set of features. However, intended future work includes (i) extending the study by exploring other related consensus and correlation techniques (*e.g.*, voting-based consensus and absolute majority principles for correlation), as well as (ii) studying other machine learning models with a more selective set of features for observation under simplicity criteria. All these improvement objectives will be carried out in (iii) real scenarios that allow demonstrating the effectiveness of the approach for the benefit of the end user. In the future, we intend to continue exploring this topic by examining the capabilities that new approaches to AI and agents can bring to the field of

charging infrastructures and their limited subsystems, including their implications.

CRediT authorship contribution statement

Cristina Alcaraz: Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Javier Lopez:** Writing – review & editing, Investigation, Funding acquisition, Formal analysis, Conceptualization. **Alberto Garcia:** Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: J. Lopez reports financial support was provided by EC. C. Alcaraz: Given her role as Associate Editor, had not involvement in the peer review of this article and had no access to information regarding its peer review. Full responsibility for the editorial process for this article was delegated to another journal editor. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has partially been supported by the project AIAS funded by the European Commission (EC) under Grant 101131292 (HORIZON-MSCA-2022-SE-01), SYNAPSE funded by the EC under Grant 101120853 (HORIZON-CL3-2022-CS-01), and DUCA funded by the EC under Grant 101086308 (HORIZON-MSCA-2021-SE-01). Additionally, this paper has received funding for open access charge by Universidad de Málaga/CBUA.

Data availability

The authors do not have permission to share data.

References

[1] IEA, Global electric car sales exceeded 17 million in 2024. https://www.iea.org/reports/global-ev-outlook-2025.
[2] M. Research, EV Charging Station Market worth $76.31 billion by 2032, 2026, https://www.marketsandmarkets.com/PressReleases/electric-vehicle-supply-equipment.asp.
[3] IEA, Global EV Outlook 2021, 2021, https://www.iea.org/reports/global-ev-outlook-2021.
[4] O.C. Alliance, OCPP 2.0.1 (Part 0-4), 2020, https://www.openchargealliance.org.
[5] C. Alcaraz, J. Cumplido, A. Trivino, OCPP in the spotlight: threats and countermeasures for electric vehicle charging infrastructures 4.0, Int. J. Inf. Secur. (2023) 1–27, http://dx.doi.org/10.1007/s10207-023-00698-8.
[6] A. Kaur, N. Valizadeh, D.N. Jha, T. Szydlo, J. R. K. Rajasekaran, V. Kumar, M. Barika, J. Liang, R. Ranjan, O. Rana, Cybersecurity Challenges in the EV Charging Ecosystem, ACM Comput. Surv. 58 (1) (2025) http://dx.doi.org/10.1145/3735662.
[7] Z. Garofalaki, D. Kosmanos, S. Moschoyiannis, D. Kallergis, C. Douligeris, Electric vehicle charging: A survey on the security issues and challenges of the open charge point protocol (OCPP), IEEE Commun. Surv. Tutorials 24 (3) (2022) 1504–1533, http://dx.doi.org/10.1109/COMST.2022.3184448.
[8] C. Alcaraz, J. Lopez, S. Wolthunsen, OCPP Protocol: Security Threats and Challenges, IEEE Trans. Smart Grid 8 (2017) 2452–2459, http://dx.doi.org/10.1109/TSG.2017.2669647.
[9] C. Alcaraz, C. Fernandez-Gago, J. Lopez, An early warning system based on reputation for energy control systems, IEEE Trans. Smart Grid 2 (4) (2011) 827–834, http://dx.doi.org/10.1109/TSG.2011.2161498.
[10] R. Gottumukkala, R. Merchant, A. Tauzin, K. Leon, A. Roche, P. Darby, Cyber-physical system security of vehicle charging stations, in: 2019 IEEE Green Technologies Conference (GreenTech), IEEE, 2019, pp. 1–5.
[11] S. Köhler, R. Baker, M. Strohmeier, I. Martinovic, Brokenwire : Wireless Disruption of CCS Electric Vehicle Charging, 2022, arXiv:2202.02104.
[12] O.G.M. Khan, E. El-Saadany, A. Youssef, M. Shaaban, Impact of electric vehicles botnets on the power grid, in: IEEE Electrical Power and Energy Conference, IEEE, 2019, pp. 1–5.
[13] A. Brighente, M. Conti, D. Donadel, F. Turrin, EVScout2.0: Electric vehicle profiling through charging profile, 2021, arXiv preprint arXiv:2106.16016.
[14] C. Alcaraz, J. Lopez, WASAM: A Dynamic Wide-Area Situational Awareness Model for Critical Domains in Smart Grids, Future Gener. Comput. Syst. 30 (2014) 146–154, http://dx.doi.org/10.1016/j.future.2013.06.030.
[15] C. Alcaraz, J. Lopez, Wide-area situational awareness for critical infrastructure protection, Computer 46 (4) (2013) 30–37, http://dx.doi.org/10.1109/MC.2013.72.
[16] J.E. Rubio, R. Roman, C. Alcaraz, Y. Zhang, Tracking Advanced Persistent Threats in Critical Infrastructures through Opinion Dynamics, in: European Symposium on Research in Computer Security (ESORICS 2018), vol. 11098, Springer, Barcelona, Spain, 2018, pp. 555–574, http://dx.doi.org/10.1007/978-3-319-99073-6_27, URL https://link.springer.com/chapter/10.1007/978-3-319-99073-6_27.
[17] M. Pagani, W. Korosec, N. Chokani, R.S. Abhari, User behaviour and electric vehicle charging infrastructure: An agent-based model assessment, Appl. Energy 254 (2019) 113680.
[18] A. Vijayashankar, Modeling Electric Vehicle Charging Infrastructure Deployment and Usage with an Agent-Based Approach, Master thesis, TU Eindhoven, 2017.
[19] E.L. Karfopoulos, N.D. Hatziargyriou, A multi-agent system for controlled charging of a large population of electric vehicles, IEEE Trans. Power Syst. 28 (2) (2012) 1196–1204.
[20] J. Miranda, J. Borges, D. Valério, M.J. Mendes, Multi-agent management system for electric vehicle charging, Int. Trans. Electr. Energy Syst. 25 (5) (2015) 770–788.
[21] S. Mocci, N. Natale, F. Pilo, S. Ruggeri, Multi-agent control system to coordinate optimal electric vehicles charging and demand response actions in active distribution networks, 2014.
[22] S. Kamboj, W. Kempton, K.S. Decker, Deploying power grid-integrated electric vehicles as a multi-agent system, in: The 10th International Conference on Autonomous Agents and Multiagent Systems, Vol. 1, 2011, pp. 13–20.
[23] C.B. Saner, A. Trivedi, D. Srinivasan, A cooperative hierarchical multi-agent system for EV charging scheduling in presence of multiple charging stations, IEEE Trans. Smart Grid 13 (3) (2022) 2218–2233.
[24] M. Basnet, M.H. Ali, Multi-agent deep reinforcement learning-driven mitigation of adverse effects of cyber-attacks on electric vehicle charging station, 2022, arXiv preprint arXiv:2207.07041.
[25] M.F. Fard, X. Huo, M. Liu, Exploration of For-Purpose Decentralized Algorithmic Cyber Attacks in EV Charging Control, in: 2023 IEEE 32nd International Symposium on Industrial Electronics, ISIE, IEEE, 2023, pp. 1–6.
[26] D.D. Sharma, S. Singh, J. Lin, E. Foruzan, Agent-based distributed control schemes for distributed energy storage systems under cyber attacks, IEEE J. Emerg. Sel. Top. Circuits Syst. 7 (2) (2017) 307–318.
[27] W.M. Stout, Toward a multi-agent system architecture for insight & cybersecurity in cyber-physical networks, in: 2018 International Carnahan Conference on Security Technology, ICCST, IEEE, 2018, pp. 1–5.
[28] M. Panfili, A. Giuseppi, A. Fiaschetti, H.B. Al-Jibreen, A. Pietrabissa, F.D. Priscoli, A game-theoretical approach to cyber-security of critical infrastructures based on multi-agent reinforcement learning, in: 2018 26th Mediterranean Conference on Control and Automation, MED, IEEE, 2018, pp. 460–465.
[29] T. Zhou, K. Xiahou, L. Zhang, Q. Wu, Multi-agent-based hierarchical detection and mitigation of cyber attacks in power systems, Int. J. Electr. Power Energy Syst. 125 (2021) 106516.
[30] M. Rajaee, K. Mazlumi, Multi-agent distributed deep learning algorithm to detect cyber-attacks in distance relays, IEEE Access 11 (2023) 10842–10849.
[31] R. Sepehrzad, M.J. Faraji, A. Al-Durra, M.S. Sadabadi, Enhancing cyber-resilience in electric vehicle charging stations: A multi-agent deep reinforcement learning approach, IEEE Trans. Intell. Transp. Syst. 25 (11) (2024) 18049–18062, http://dx.doi.org/10.1109/TITS.2024.3408238.
[32] R. Honnalli, J. Farooq, LLM-Powered Agentic AI Approach to Securing EV Charging Systems Against Cyber Threats, in: 2025 IEEE 26th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2025, pp. 266–274, http://dx.doi.org/10.1109/WoWMoM65615.2025.00053.
[33] M. Alauthman, A. Aldweesh, A. Al-Qerem, A. Al Maqousi, Reinforcement Learning-Based Intrusion Detection for Electric Vehicle Charging Stations, in: 2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications, ACDSA, 2025, pp. 1–5, http://dx.doi.org/10.1109/ACDSA65407.2025.11165835.
[34] C. Alcaraz, J. Lopez, Digital twin-assisted anomaly detection for industrial scenarios, Int. J. Crit. Infrastruct. Prot. 47 (2024) 100721, http://dx.doi.org/10.1016/j.ijcip.2024.100721, URL https://www.sciencedirect.com/science/article/pii/S1874548224000623.
[35] O.C. Alliance, Open Charge Point Protocol, 2023, https://www.openchargealliance.org.

[36] C. Alcaraz, J.E. Rubio, J. Lopez, Blockchain-Assisted Access for Federated Smart Grid Domains: Coupling and Features, J. Parallel Distrib. Comput. 144 (2020) 124–135.

[37] P. Karn, C. Partridge, Improving round-trip time estimates in reliable transport protocols, ACM SIGCOMM Comput. Commun. Rev. 17 (5) (1987) 2–7.

[38] S. Zander, G. Armitage, P. Branch, Covert channels in the IP time to live field, in: Australian Telecommunication Networks and Application Conference, (ATNAC) 2006, 2006.

[39] L. Tamilselvan, V. Sankaranarayanan, Prevention of Blackhole Attack in MANET, in: The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007), 2007, p. 21, http://dx.doi.org/10.1109/AUSWIRELESS.2007.61.

[40] Z. Fan, X. Zi-xuan, W. Ming-hu, Fault diagnosis method for lithium-ion batteries in electric vehicles using generalized dimensionless indicator and local outlier factor, J. Energy Storage 52 (2022) 104963, http://dx.doi.org/10.1016/j.est.2022.104963, URL https://www.sciencedirect.com/science/article/pii/S2352152X22009690.

[41] Q. Feng, H. Li, Y. Zhou, D. Feng, Y. Wang, Y. Su, Review of electric vehicles' charging data anomaly detection based on deep learning, in: 2022 Power System and Green Energy Conference, PSGEC, 2022, pp. 337–341, http://dx.doi.org/10.1109/PSGEC54663.2022.9881073.

[42] D. Velasquez, E. Perez, X. Oregui, A. Artetxe, J. Manteca, J.E. Mansilla, M. Toro, M. Maiza, B. Sierra, A Hybrid Machine-Learning Ensemble for Anomaly Detection in Real-Time Industry 4.0 Systems, IEEE Access 10 (2022) 72024–72036.

[43] Y. Peng, Y. Yang, Y. Xu, Y. Xue, R. Song, J. Kang, H. Zhao, Electricity theft detection in AMI based on clustering and local outlier factor, IEEE Access 9 (2021) 107250–107259.

[44] R. Hegselmann, U. Krause, et al., Opinion dynamics and bounded confidence models, analysis, and simulation, J. Artif. Soc. Soc. Simul. 5 (3) (2002).