Wireless Networks
https://doi.org/10.1007/s11276-023-03375-9

A blockchain ledger for securing isolated ambient intelligence deployments using reputation and information theory metrics

Borja Bordel[1] · Ramón Alcarria[1] · Tomás Robles[1]

Accepted: 2 May 2023

Abstract

Ambient Intelligence deployments are very vulnerable to Cyber-Physical attacks. In these attacking strategies, intruders try to manipulate the behavior of the global system by affecting some key elements within the deployment. Typically, attackers inject false information, integrate malicious devices within the deployment, or infect communications among sensor nodes, among other possibilities. To protect Ambient Intelligence deployments against these attacks, complex data analysis algorithms are usually employed in the cloud to remove anomalous information from historical series. However, this approach presents two main problems. First, it requires all Ambient Intelligence systems to be networked and connected to the cloud. But most new applications for Ambient Intelligence are supported by isolated systems. And second, they are computationally heavy and not compatible with new decentralized architectures. Therefore, in this paper we propose a new decentralized security solution, based on a Blockchain ledger, to protect isolated Ambient Intelligence deployments. In this ledger, new sensing data are considered transactions that must be validated by edge managers, which operate a Blockchain network. This validation is based on reputation metrics evaluated by sensor nodes using historical network data and identity parameters. Through information theory, the coherence of all transactions with the behavior of the historical deployment is also analyzed and considered in the validation algorithm. The relevance of edge managers in the Blockchain network is also weighted considering the knowledge they have about the deployment. An experimental validation, supported by simulation tools and scenarios, is also described. Results show that up to 93% of Cyber-Physical attacks are correctly detected and stopped, with a maximum delay of 37 s.

Keywords Blockchain · Ambient intelligence · Reputation · Information theory · Consensus protocols · Data security

1 Introduction

In the last ten years, many new technological paradigms have been described, developed, and (finally) applied to a large catalog of different scenarios. From Cyber-Physical Systems [1] and Ambient Intelligence [2] to Industry 4.0 [3] and the Industrial Internet of Things [4]. But together with this technological revolution, a new family of digital risks and vulnerabilities has emerged. Among all these innovative attacking strategies, Cyber-Physical attacks are probably the most dangerous and worrying.

In Cyber-Physical attacks [5], intruders take advantage of feedback control loops and other similar algorithms deployed within technological platforms, to amplify and extend the impact of their attack strategy across the entire system architecture. Only a very specific (and typically small) manipulation or malicious action over the key (vulnerable) element or component is necessary to affect and modify the behavior of the whole system. Although that key vulnerable element would be different for each attack, deployment, technology, and implementation, or event it could not exist, some technological paradigms and architecture allow identifying those components which could be vulnerable with a higher probability.

✉ Borja Bordel
borja.bordel@upm.es

Ramón Alcarria
ramon.alcarria@upm.es

Tomás Robles
tomas.robles@upm.es

[1] Universidad Politécnica de Madrid, Campus Sur. Ctra. de Valencia, Km. 7, Madrid, España

Published online: 24 May 2023

Actually, among all innovative technological paradigms, Ambient Intelligence (AmI) is where more clearly the potential attacking vectors for a Cyber-Physical attack can be identified. Every AmI deployment is supported by a large and dense network of sensor nodes, including thousands of resource-constrained devices [6]. To make the deployment and management of such a large network feasible, devices are randomly distributed, connected, or replaced [7]. It is not a planned network, and there is no exhaustive description of its structure or the participating devices. In this context, sensor nodes must be able to self-configure and start data capture and transmission automatically; and any device operating with the proper configuration and sending information to the correct endpoint is accepted as part of the AmI deployment [8].

For Cyber-Physical attackers, therefore, it is easy to inject false information into the AmI hardware platform, through intruder malicious devices or infecting the communications among legitimate sensor nodes (among other possibilities) [9]. To protect AmI deployments against these attacks, complex data analysis algorithms are usually employed. Using stochastics models [10], time series [11] and artificial intelligence [12], false information can be detected, corrected, and/or removed. But these algorithms are too computationally heavy to be maintained by sensor nodes and are usually deployed in the cloud. However, this approach presents two main problems.

First, it requires all Ambient Intelligence systems to be networked and connected to the cloud. And, although some AmI applications are actually networked, in most scenarios, AmI deployments are isolated [7]. Either because the system is deployed in a remote location without communication infrastructure (such as in natural environment monitoring application or digital agriculture solutions), or because we are dealing with a critical system where Internet connections are not allowed (such as in many Industry 4.0 applications and military missions). Many AmI systems do the data processing locally, with no option to execute complex algorithms for false information mitigation.

And second, this protection strategy requires centralized data management. Stochastic models or artificial intelligence algorithms need a full vision of the data captured and accepted within the AmI deployment to be precise. Thus, all information must be transmitted and accumulated at the same point. But this centralized scheme is very slow (because of transmission delays) and computationally heavy. Mostly, this approach is not compatible with the most innovative decentralized architectures, such as edge computing [13]. In these architectures, data processing tends to be decomposed into atomic tasks that can be delegated and solved only one step away from the sensor networks (a layer known as "edge"). As a result, AmI deployments would reduce their reaction capacity, performance, and the catalog of applications they can cover.

In conclusion, new distributed solutions are needed to protect isolated AmI deployments against false information injections caused by Cyber-Physical attacks. These solutions should not be computationally heavy, so they can be supported by sensor nodes and edge devices.

Therefore, in this paper we propose a new decentralized security solution, based on a Blockchain ledger. In this ledger, new sensing data are considered transactions that must be validated by edge managers, which operate a Blockchain network. This validation is based on two different information sources. On the one hand, sensor nodes are validated as valid information sources through reputation metrics. These metrics combine implicit reputation indicators computed by sensor nodes using historical network data and explicit reputation indicators based on identity parameters. On the other hand, new data are evaluated to verify whether they are potentially legitimate or not. Information theory metrics and the coherence of new transactions with the behavior of the historical deployment are analyzed to evaluate their legitimacy. Only validated transactions are considered, stored, and processed.

Edge managers maintain the Blockchain network and execute the validation algorithm. As they do not have a full vision of the AmI deployment and different edge managers collect information from different groups of sensor nodes, validation must be achieved through consensus. The consensus protocol that we propose requires a weighted majority of edge managers to validate a block to finally be accepted. For this "voting", the relevance of edge managers in the Blockchain network is weighted considering the knowledge they have about the deployment, and the historical series of previously accepted (and rejected) blocks.

The remainder of the paper is organized as follows. Section 2 discusses the state of the art in security and protection mechanisms for AmI deployments. Section 3 presents the proposed security solution, including the validation algorithm and the consensus protocol. Section 4 describes the experimental methodology and analyzes the obtained results. Section 5 concludes the paper.

2 State of the art on AmI security solutions

Currently, the most discussed and popular topic with respect to AmI security is governance [37]. In general, the social and administrative dimensions of AmI security are deeply analyzed, from the acceptance and perception of privacy in AmI systems [38] to techniques to make AmI technologies accountable and responsible (in legal terms) [39].

However, traditional and still open security risks and vulnerabilities in AmI systems have also been exhaustively studied. Most reported security solutions for Ambient Intelligence systems are designed for deployments with Internet connection and communicating to the cloud. In this context, risks associated with public networks are dominant and well-known security technologies are studied, such as HTTPS (HyperText Transfer Protocol Secure) protocol and certificates [14] or cryptography based on elliptic curves [15]. Innovative security schemes for these networked AmI systems may also be found, although they are sparse. In this category, Intrusion Detection Systems (IDS) for Ambient Intelligence are the most popular approach. Some authors propose honeypots to capture information about attackers and feed a severity analyzer supported by reinforcement learning algorithms [16]. Other works employ advanced access control policies to identify intruding devices, for example, using optimization problems, convex functions, and dual decompositions [17] to model and handle the wireless network. On the other hand, for those AmI deployments where web protocols are implemented, trustworthiness is also studied and enhanced. For example, through trusted ontology frameworks, which can be adapted and personalized to the specific services provided by each different AmI system [18]. Finally, some authors propose centralized data repositories, so general stochastic models [10] can be applied to detect outliers, incoherent information, and, eventually, Cyber-Physical attacks. Although these algorithms aim to correct and clean stored data from malicious samples, intrusion detection is just a secondary, very limited application.

However, none of these approaches is adequate for isolated AmI deployments. In fact, in isolated AmI deployments, security mechanisms must be supported by sensor nodes and edge managers. And in this context, low-level network parameters are typically employed to monitor and control intruders. In standard mesh networks, parameters such as reliability are periodically considered and updated to make decisions about which devices remain connected and which ones are blacklisted and removed [20]. However, in layered networks, such as Publication/Subscription networks, edge devices must implement analysis algorithms [19] (using, for example, artificial intelligence) as they only get indirect observations about the sensor nodes and their behavior. The main problem of all these solutions is their low precision. Both the false negative and false positive rates usually grow up linearly with the number of sensor nodes, as errors are accumulative, and network parameters are calculated aggregating information from all devices within the network. Although errors are below 2% for small networks (less than fifty nodes), they increase above 20% for large deployments (more than a thousand devices). That is not acceptable for most AmI applications.

To mitigate this situation and improve the performance of low-level security mechanisms, some works propose improved indicators representing “trust” in AmI networks, but with much lower associated errors. To do that, they combine network information with social information [21]. However, the results show only that these indicators are more stable and present a lower calculation error than previous proposals. And there is no information on how they would behave when integrated into a real security solution and AmI deployment. In conclusion, AmI security remains an open challenge, and very recent work [22] confirms this conclusion by describing all pending issues within this research topic.

In this context, most recent proposals work in two different directions. On the one hand, as attack detection is a complex task, some authors propose frameworks to detect the most vulnerable components within an AmI deployment [9]. The final objective is to correct or mitigate all these vulnerabilities, but (sometimes) the state of the art does not allow for it. Like it happens with the very novel Cyber-Physical attacks. On the other hand, security mechanisms based on Blockchain networks have been reported.

Several authors have confirmed the benefits of Blockchain technologies when applied to AmI deployments [23]. And although several works on unions between AmI systems and Blockchain have been reported [24], most of them require sensor nodes to communicate with the global Internet and the cloud. In the most common approach, Blockchain networks are independent of the AmI deployment and operate in the cloud. For example, new secure access control protocols in which Blockchain networks are supported by edge-cloud collaboration [35]. Or authentication services for smart homes, where a Metropolitan Area Network (MAN) is required connecting different regions of the city to communicate with the Blockchain provider of the region [36]. Some works use Blockchain as a public, transparent, and reliable registration system for sensor nodes [25]. In this solution, other peer nodes and manager devices use transparent information to determine if nodes are legitimate or malicious. Furthermore, some authors describe mechanisms in which AmI data are collected through public general Blockchain networks and instruments, such as the InterPlanetary File System (IPFS) [26].

Public Blockchain networks (such as Ethereum) are also used to support automatic alert and incidence management in AmI systems [27], although in this case Blockchain is more related to automation than to security. Similarly, Blockchain for AmI networks has been used as a communication system [28] or a payment platform [29].

However, again, all of these solutions are designed for AmI deployments connected to the cloud. It is difficult to find Blockchain-based technologies specifically designed for isolated AmI deployments. Works describing new consensus protocols (based on network indicators), so that Blockchain can be executed by sensor nodes have been reported [30]. But results show sensor nodes are not powerful enough to maintain a Blockchain network in the long term [31] (most of the computing time is consumed by the Blockchain protocols), and no performance analysis of these new protocols when integrated into AmI systems have been reported. In fact, existing results assume AmI deployments have Internet connection [32], so isolated deployments are not studied.

Our paper aims to fill this gap. In this paper we describe a new Blockchain ledger, but to be supported by edge managers so it is computationally sustainable at long-term. It includes a new consensus protocol adapted to isolated AmI deployments, where no connection to the cloud is needed. We use network parameters, together with other information sources and analysis algorithms, to improve the precision and success rate reported in the state of the art, correctly detecting up to 93% of Cyber-Physical attacks.

3 A new blockchain ledger for AmI securization

Isolated AmI deployments are supported by a three-layer architecture (see Fig. 1). The first layer is composed of randomly distributed sensor nodes that capture data on a random basis. In this paper we are assuming N sensor nodes n_i (1) are part of this AmI deployment. In the second layer, M edge managers m_i (2) communicate with the K_i sensor nodes $n_j^{\mathcal{C}_i}$ within their coverage area $\mathcal{C}_i$ (3). Coverage areas are not homogeneous and are unknown a priori, as they become self-configured when the AmI deployment starts operating. The third layer is composed of a local data processing server (usually distributed), whose internal structure and behavior are transparent for the purpose of this paper.

$$\mathcal{N} = \{n_i i = 1, \ldots, N\} \tag{1}$$

$$\mathcal{M} = \{m_i i = 1, \ldots, M\} \tag{2}$$

$$\mathcal{C}_i = \left\{ n_j^{\mathcal{C}_i} j = 1, \ldots, K_i \right\} \tag{3}$$

In this architecture, edge managers $\mathcal{M}$ maintain a data structure in a collaborative way. It is a Blockchain C (4) (hereinafter referred to "chain" too), that is, a sequence of connected blocks b_i where each new block $C[i+1]$ contains an explicit reference to the previous one $C[i]$ through its hash. Blocks b_i contain a random number T_i of accepted transitions t_j^i (or operations), invoked by sensor nodes $\mathcal{N}$ (5). In our proposal, these transactions t_j^i represent the transmission of new data that are accepted by edge managers as legitimate.

$$C = \{b_i\} = \{C[i]\} being b_i \equiv C[i] \tag{4}$$

$$b_i = \left\{ t_j^i j = 1, \ldots, T_i \right\} \tag{5}$$

A ledger is a set of mechanisms, shared and common to all edge managers $\mathcal{M}$, employed to maintain updated and coherent the chain and the list of accepted transactions, according to the common acceptance criteria and guaranteeing the consensus among all the edge managers. In this paper, we propose a Blockchain ledger where three basic mechanisms are considered. First, a reputation model and calculation framework, including explicit and implicit reputation indicators. This mechanism is used to identify legitimate data sources whose transactions may eventually be included in the ledger (see Sect. 3.1). Second, a stochastic framework to calculate how probable a new data is to be legitimate. In this framework, probabilities are obtained using information theory indicators and considering the historical behavior of the AmI deployment (see Sect. 3.2). This instrument is used to identify valid transactions. And third, and finally, a new consensus protocol and block generation and transaction validation algorithms. Considering reputation indicators and data validity probabilities, these instruments identify fully valid transactions through weighted consensus among all edge managers (see Sect. 3.3).

3.1 Sensor node validation: reputation model

Reputation, as a technological parameter, can be defined using several different approaches [33]: cognitive, computational, neurological, or even game-theoretical. But in security applications, indicators must be precise and stable to avoid false positive and false negative detections. Therefore, in this paper, we propose a hybrid definition of reputation to improve the stability and precision of classic approaches [34].

In our framework, the global reputation $R[n_i]$ of sensor node n_i is obtained as the geometric average of two different reputation measures (6). On the one hand, the explicit reputation $R_e[n_i]$ obtained from direct recommendations generated by other nodes within the AmI deployment. On the other hand, the implicit reputation $R_{im}[n_i]$ calculated by the surrounding sensor nodes using traffic

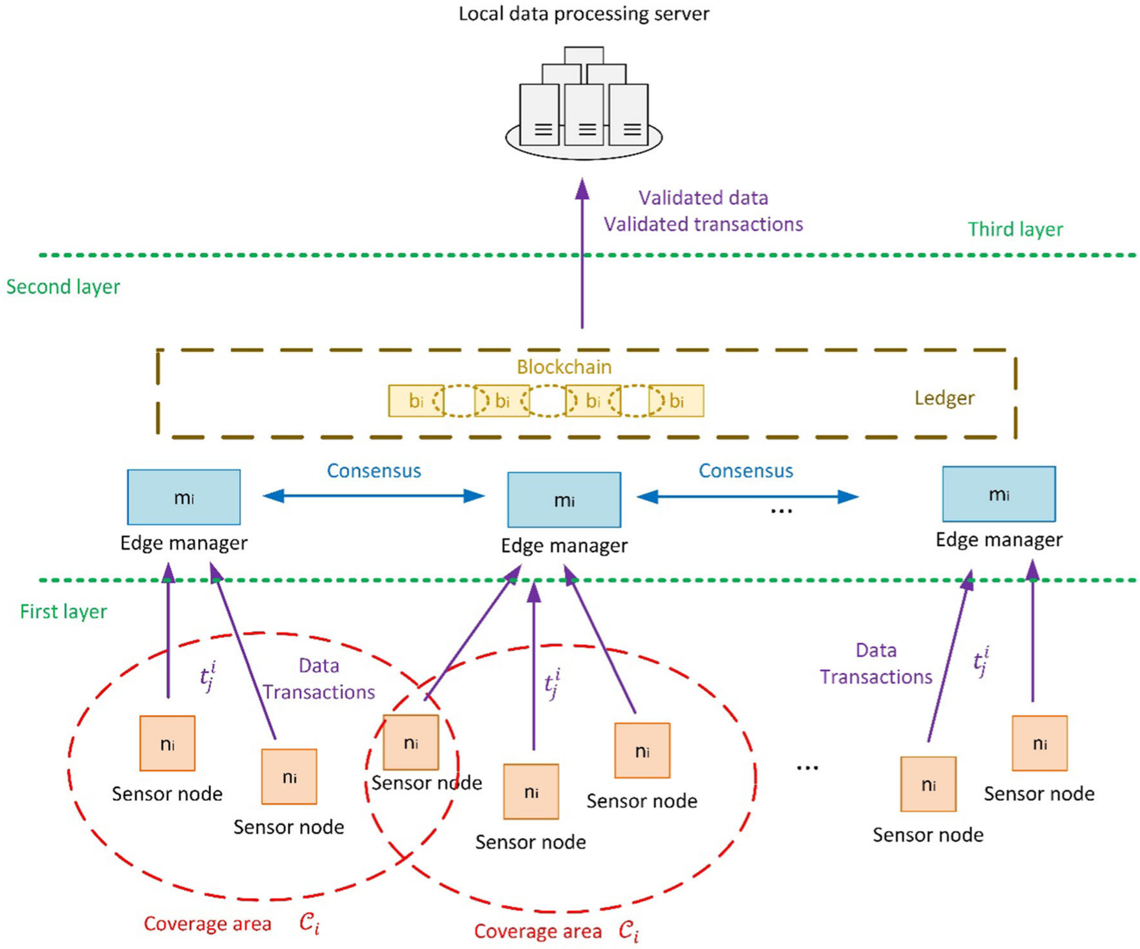


Fig. 1 Proposed architecture for an isolated AmI deployment

statistics and other network indicators. Both the implicit and the explicit reputations vary in the interval $[0, 1]$.

$$R[n_i] = \sqrt{R_e[n_i] \bullet R_{im}[n_i]} \tag{6}$$

Explicit reputation $R_e[n_i]$ is calculated by edge managers $\mathcal{M}$ from direct recommendations produced by sensor nodes n_j (being $j \neq i$). Figure 2 shows the block diagram of the proposed calculation algorithm.

Periodically, every τ_i^{en} seconds, sensor node n_i creates a map with all nodes n_j within its coverage region. For each node n_j it evaluates three identity and configuration parameters: the link address (it may be a MAC -Media Access Control- address or an UUID -Universally Unique Identifier-, for example); the communication protocols being employed; and the provided data formats and/or services. Considering the received information and responses, node n_i may generate one or several positive or negative recommendations about node n_j. Recommendations are generated according to the following criteria:

- A positive recommendation is generated if the link address belongs to a device that was part of the AmI network in the past. A negative recommendation is generated if the link address belongs to a blacklisted device.
- A positive recommendation is generated if communication protocols are standard protocols already present in the AmI deployment. And a negative recommendation is generated if protocols or configurations usually used in cyberattacks are detected.
- A positive recommendation if data formats and/or services are similar to the ones managed by other sensor nodes. A negative recommendation is produced when data formats or services are detected that are typically associated with cyberattacks.

All recommendations are sent to edge managers. When received, all recommendations are collected in two different buckets. The first one for positive recommendations and the second one for negative recommendations. Every

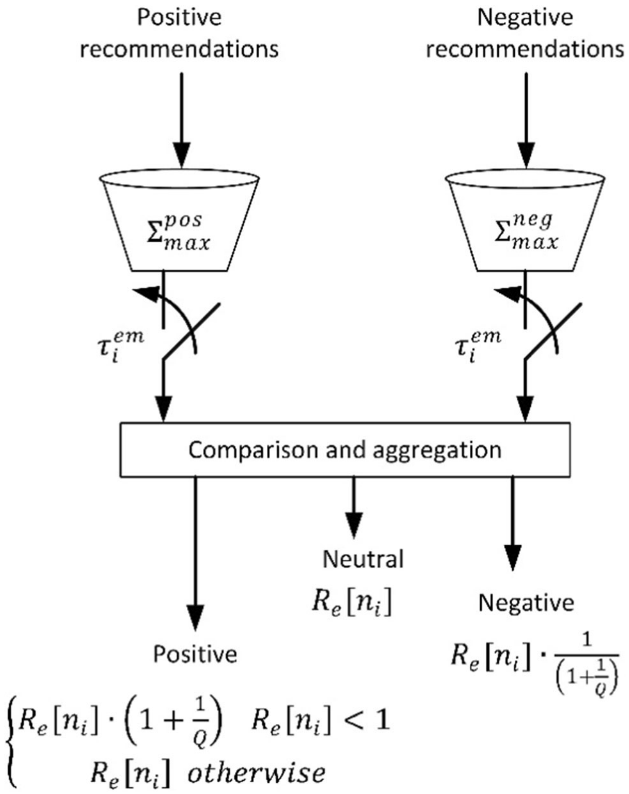
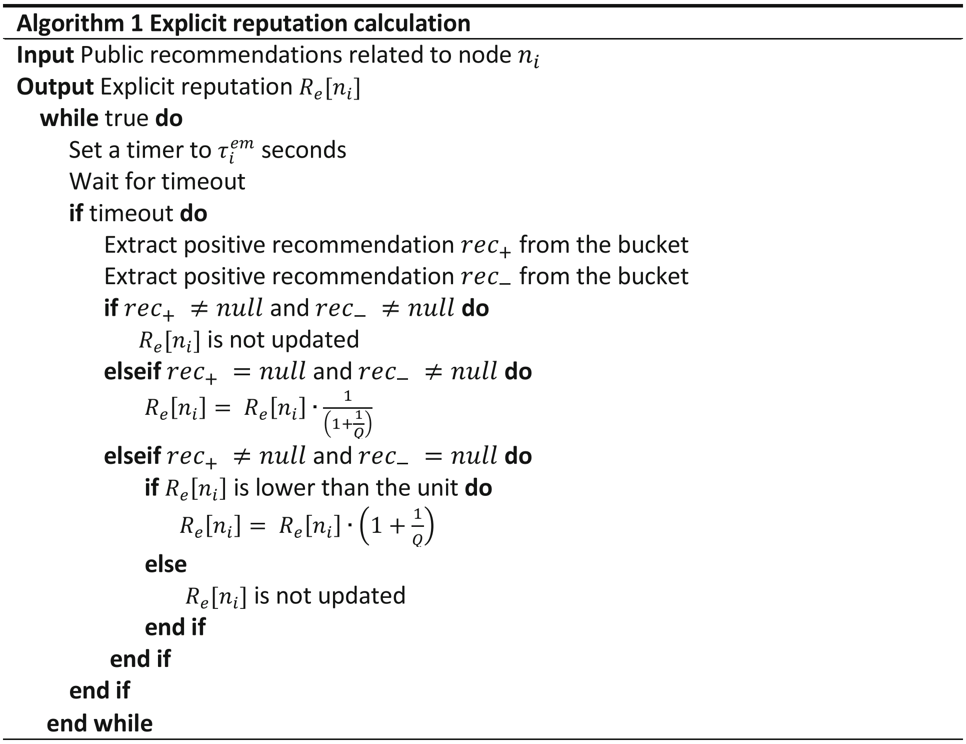

Positive recommendations
Negative recommendations
Σ_{max}^{pos}
Σ_{max}^{neg}
τ_i^{em}
τ_i^{em}
Comparison and aggregation
Neutral
$R_e[n_i]$
Negative
$R_e[n_i] \cdot \frac{1}{\left(1+\frac{1}{Q}\right)}$
Positive
$\begin{cases} R_e[n_i] \cdot \left(1+\frac{1}{Q}\right) & R_e[n_i] < 1 \\ R_e[n_i] \ otherwise \end{cases}$

Fig. 2 Explicit reputation calculation algorithm

τ_i^{em} seconds one recommendation about node n_i is extracted from both buckets. If a positive and a negative recommendation are extracted, explicit reputation remains unchanged. But if the negative bucket is empty, and only a positive recommendation is extracted, explicit reputation $R_e[n_i]$ is incremented using the multiplier $\left(1+\frac{1}{Q}\right)$ where Q is a real number higher than or equal to the unit (7). On the contrary, if the positive bucket is empty and only a negative recommendation is extracted, explicit reputation $R_e[n_i]$ is decremented using the same multiplier. This calculation procedure is described in Algorithm 1.

$$Q \in [1, \infty) \tag{7}$$

In addition, both the positive recommendation bucket and the negative recommendation bucket have a maximum capacity of Σ_{max}^{pos} and Σ_{max}^{neg} recommendations, respectively. When any bucket is full, recommendations of the corresponding type are rejected. With this design, we protect the reputation calculation framework from recommendation bursts, which are unnatural and typically associated with attacks. Besides, a correct balance between time constants τ_i^{em} and τ_i^{en} allow the algorithm to automatically compensate malicious attacks thanks to genuine recommendations coming from legitimate sensor nodes.



Algorithm 1 Explicit reputation calculation

```
Input Public recommendations related to node $n_i$
Output Explicit reputation $R_e[n_i]$
  while true do
    Set a timer to $\tau_i^{em}$ seconds
    Wait for timeout
    if timeout do
        Extract positive recommendation $rec_+$ from the bucket
        Extract positive recommendation $rec_-$ from the bucket
        if $rec_+ \neq null$ and $rec_- \neq null$ do
            $R_e[n_i]$ is not updated
        elseif $rec_+ = null$ and $rec_- \neq null$ do
            $R_e[n_i] = R_e[n_i] \cdot \frac{1}{\left(1+\frac{1}{Q}\right)}$
        elseif $rec_+ \neq null$ and $rec_- = null$ do
            if $R_e[n_i]$ is lower than the unit do
                $R_e[n_i] = R_e[n_i] \cdot \left(1+\frac{1}{Q}\right)$
            else
                $R_e[n_i]$ is not updated
            end if
        end if
    end if
  end while
```

Implicit reputation $R_{im}[n_i]$ is deducted from nodes' behavior. But every sensor node n_j has a different vision of the other nodes' behavior. Then, the global implicit reputation $R_{im}[n_i]$ is calculated by aggregating partial estimations $R_{im}(n_j)[n_i]$ generated by nodes n_j (8). However, different estimations may show different significances. Weights λ_j (9) represent these differences in the final reputation calculation.

$$R_{im}[n_i] = \sum_{\substack{\forall n_j \in N \\ j \neq i}} \lambda_j \bullet R_{im}(n_j)[n_i] \tag{8}$$

$$\lambda_j \in (0,1] \tag{9}$$

In our model, implicit reputation $R_{im}(n_j)[n_i]$ is calculated by combining three different network parameters (10):

- Reliability (ρ). It measures the availability of sensor nodes to communicate when it is requested by other nodes within the AmI deployment.
- Goodness (ζ). It represents the posteriori probability of a sensor node to become malicious and attack other nodes in its surroundings.
- Importance (v). This parameter refers to the relevance and how essential a node is within an AmI deployment. It depends, basically, on how many other nodes may assume its functions if it is removed from the system.

Each one of these network indicators is weighted (using μ_1, μ_2 and μ_3 real parameters), so it is possible to select the relative significance of each contribution to the final implicit reputation estimation (11).

$$R_{im}(n_j)[n_i] = [\,\mu_1 \quad \mu_2 \quad \mu_3\,] \bullet \begin{bmatrix} \rho \\ \zeta \\ v \end{bmatrix} \tag{10}$$

$$\mu_i \in (0,1] i = 1,2,3 \tag{11}$$

Node n_i divides time into measurement slots with a duration of τ_i^{mes} seconds. For each slot, node n_i controls the total number of communication attempts p_{total} with every node n_j within its coverage region, together with the number of attempts that were actually successful, $p_{success}$. Using the Laplace definition for probability and these two measures, we can calculate the instant availability a_{inst}^j for node n_j (12). All these instant measurements are collected in a common time series $w[k]$ (13), where k is the discrete time variable (k-th slot). Hereinafter $k = 0$ is the current (present) time instant. However, reliability depends not only on events happening in the last τ_i^{mes} seconds, but on all historical node's behavior. Although past measures are slightly less relevant than recent measurements. Thus, we can calculate the historical availability a_j of node n_j through a weighted average (14), where A is an integer parameter and weights slowly reduce their value thanks to the logarithmic function.

$$a_{inst}^j = \frac{p_{success}}{p_{total}} \tag{12}$$

$$w[k] = \left(a_{inst}^j\right) \tag{13}$$

$$a_j = \sum_{k=1}^{A} \frac{w[1-k]}{1 + ln(k)} \tag{14}$$

In network engineering, traffic is modeled as a Poisson process. Then, service time, congestion, and reliability follow an exponential distribution. In our model, reliability follows the same law (15), being K_{av} a constant controlling the equivalence between historical availability a_j and reliability ρ (16). In general, full reliability is achieved when historical availability is equal to $5 \bullet K_{av}$ or higher.

$$\rho = 1 - exp\left\{-\frac{a_j}{K_{av}}\right\} \tag{15}$$

$$K_{av} \in (0,1] \vdots a_j \geq 5 \bullet K_{av} \Rightarrow \rho \approx 1 \tag{16}$$

On the other hand, for each measurement time slot, node n_i also monitors the number of attacks it receives from node n_j. For each time slot, z_{attack}^j represents this indicator. All these instant measurements are collected in a common time series $x[k]$ (17), where k is the discrete time variable (k-th slot).

$$x[k] = \left(z_{attack}^j\right) \tag{17}$$

As before, this is an instant value; however, reputation does not only depend on the events that occurred during the last time slot, but also on the entire historical behavior. However, past behaviors are not as relevant as recent events. And, regarding attacks, fast adaptation to malicious behaviors allows quick detection and mitigation. Thus, a global historical attack counter z_j is obtained through a weighted average (18), where weights follow an exponential law which reduces the significance of past behaviors much faster than logarithmic laws. Being Z and r integer parameters higher than the unit.

$$z_j = \sum_{k=0}^{Z} x[-k] \bullet \left(\frac{1}{r}\right)^{k+1} \tag{18}$$

Using this global indicator, and through a sigmoid function (19), we calculate the goodness ζ of node n_j.

$$\zeta = \frac{1}{z_j \bullet \sqrt{1 + \left(\frac{1}{z_j}\right)^2}} \tag{19}$$

Finally, node n_i may calculate the importance of n_j using two parameters. First, parameter cl indicates how critical are services or data provided by node n_j in the context of the AmI deployment. Parameter cl takes values in the interval $(0,\infty]$, where lower values indicate the component's criticality is limited. This parameter must be defined by AmI system managers and cannot be self-selected by sensor nodes. Secondly, parameter red represents the network redundancy. It measures how many other nodes within the coverage region of node n_i are providing the same services or data than node n_j. It can be estimated using the cardinality $card\{\bullet\}$ operator (20).

$$red = \frac{card\{similarnodesn_j\}}{card\{totaln_jnodes\}} \tag{20}$$

Using these two parameters, importance can be calculated (21). As can be seen, through the proposed exponential law, importance v decreases as red parameter increases, but the decreasing rate is slower as the parameter cl is higher.

$$v = exp\{-red \bullet cl\} \tag{21}$$

In this case, instant (last) calculation is the only relevant, and historical values for importance v are not considered. Configurations of the past deployment do not affect the performance of the current AmI deployment.

Finally, for validation purposes, any sensor node n_i is considered to have a good reputation and then, validated as a legitimate information source when its reputation $R[n_i]$ goes above a threshold R_{th} (22).

$$ifR[n_i] \geq R_{th} \Rightarrow n_ivalidated \tag{22}$$

3.2 Data validation

Even valid and legitimate nodes may be infected or affected by malicious effects, making them to generate false information. Then, in addition to the reputation calculation framework, a mechanism for data validation is essential.

In our model, each transaction or data sample is characterized by a Bernoulli distribution $\mathcal{B}(q)$ where transaction t_j^i is false with probability q, and it is legitimate with probability $(1-q)$ (23).

$$\mathcal{B}(q) \sim P\left(t_j^i \; is \; valid\right) = \begin{cases} 1-q & if \; true \\ q & if \; false \end{cases} \tag{23}$$

Every transactiont_j^i, then, is fully characterized by probability$q$. This probability is obtained by combining three stochastic indicators. First, information entropy,$H(\bullet)$. In general, legitimate data series have a low entropy because they follow a deterministic pattern. High entropy is evidence of malicious attack trying to confuse the AmI system. Second, mutual information$I(\bullet;\bullet)$. Close and similar nodes are expected to generate data series sharing a large amount of information. A mid-term or long-term sequence of outliers is evidence of the injection of false information. And, finally, the Probability Density Function (PDF),$f(\bullet)$. Physical processes are typically stationary, and probability distribution of data samples does not change with time. Radical, fast, or unexpected changes in the probability distribution of data samples are evidence of false information too.

However, all these stochastic indicators cannot be applied to individual transactions t_j^i but a sequence or collection T_L including L transactions. Then, edge managers must collect L transactions and, later, validate all together through the same data validation process. Parameter L has not to be fixed and may change with time according to the AmI deployment's needs. Hereinafter, $y_i[k]$ is the series with all the history of data generated by node n_i (as transactions t_j^i), and $y_i^L[k]$ is the series with the last L samples generated by node n_i (contained in collection T_L).

Information entropy, $H(\bullet)$, is directly applied to series $y_i^L[k]$ (24), where p is the probability of symbol (or data sample) within the series $y_i^L[k]$. In order to calculate probabilities p the Laplace's definition for probability is employed (25). But because of noise, fluctuations, numerical errors, etc., sensor nodes rarely generate two identical samples. Then, in our model all samples within the range $[-\varepsilon, +\varepsilon]$ are considered to be same, where ε is a real parameter representing the precision (absolute value) of the AmI system. This operation is performed by the Heaviside's step function, $u[\bullet]$.

$$H\left(y_i^L\right) = -\sum_{\forall differenty \in y_i^L} p_y \bullet \log_2\left(p_y\right) \tag{24}$$

$$p_y = \frac{1}{L} \bullet \sum_{\forall \tilde{y} \in y_i^L} u[\varepsilon - |y - \tilde{y}|] \tag{25}$$

Besides, information entropy varies in the range $[0, s_i]$ (in bits), being s is the exponent satisfying an exponential equality (26) where S_i^t is the number of different symbols (data samples) in the sequence $y_i^L[k]$. But, for coherence with the other indicators, it is convenient if entropy H also varies in the interval $[0,1]$ (as probability functions do). Then, a mapping function is applied to get the final value for entropy H_{map} (27).

$$S_i^t = 2^{s_i} \tag{26}$$

$$H_{map}\left(y_i^L\right) = H\left(y_i^L\right) \bullet \frac{1}{s_i} \tag{27}$$

On the other hand, mutual information $I(\bullet;\bullet)$ is applied to series $y_i^L[k]$ and $y_j^L[k]$ (28), being n_i and n_j equivalent or similar nodes, according to explicit reputation parameters previously described in Sect. 3.1. p_{1} is the probability of symbol $_1$ (or data sample) within the series$y_i^L[k]$, p_{2} is the probability of symbol $_2$ (or data sample) within the series$y_j^L[k]$, and being $p_{1,2}$ the joint probability of symbols (or data samples) $_1$ and $_2$ to be generated at the same time instant by nodes n_i and n_j respectively. As before, in order to calculate probabilities p_{1} (29), p_{2} (30) and $p_{1,2}$ (31) the Laplace's definition for probability is employed and being $u[\bullet]$ the Heaviside's step function. Additionally, for the mutual information calculation we are considering a tolerance rangeε, so two samples are assumed to be the same if their difference is lower than this tolerance. Equally, sensor nodes in AmI deployments are not synchronized. Thus, all samples within a given time tolerance π (discrete time units) are considered to be generated at the same instant.

$$I\left(y_i^L; y_j^L\right) = \sum_{\forall different_1 \in y_i^L} \sum_{\forall different_2 \in y_j^L} p_{1,2} \bullet \log_2\left(\frac{p_{1,2}}{p_{1} \bullet p_{2}}\right) \tag{28}$$

$$p_{y_1} = \frac{1}{L} \bullet \sum_{\forall \tilde{y} \in y_i^L} u[\varepsilon - |y_1 - \tilde{y}|] \tag{29}$$

$$p_{y_2} = \frac{1}{L} \bullet \sum_{\forall\ \tilde{y} \in y_j^L} u[\varepsilon - |y_2 - \tilde{y}|] \tag{30}$$

$$p_{1,2} = \frac{1}{L \bullet (2\pi + 1)} \bullet \sum_{k=-L+1}^{0} \sum_{k_0=-\pi}^{\pi} \left(u\left[\varepsilon - \left|_2 - y_j^L[k - k_0]\right|\right] \bullet u\left[\varepsilon - \left|_1 - y_i^L[k]\right|\right]\right) \tag{31}$$

In this case, mutual information also varies in the interval $[0, s_{i,j}]$, where $s_{i,j}$ is the solution to the exponential Eq. (32) to calculate the total number of different samples (symbols) $S_{i,j}^t$ in series $y_i^L[k]$ and $y_j^L[k]$. Again, we employ a mapping function (33) to move the target interval to the range $[0, 1]$ (where probability functions usually take values) and obtain the final mutual information I_{map}.

$$S_{i,j}^t = 2^{s_{i,j}} \tag{32}$$

$$I_{map}\left(y_i^L; y_j^L\right) = I\left(y_i^L; y_j^L\right) \bullet \frac{1}{s_{i,j}} \tag{33}$$

And third, and finally, the Probability Density Function (PDF) allows to analyze the occurrence probability of every individual sample or transaction t_j^i. To calculate the PDF $f(\bullet)$, we use the entire series $y_i[k]$, the Heaviside's step function, $u[\bullet]$, the cardinality operator $card\{\bullet\}$, and the probability theory (34), so histograms approach to the PDF when the number of realizations is high. As before, all samples within the t $[-\varepsilon, +\varepsilon]$ are considered to be equal, where ε is an integer value representing the precision of the AmI system.

$$f(y) = \frac{1}{card\{y_i\}} \bullet \sum_{\forall \tilde{y} \in y_i} u[\varepsilon - |y - \tilde{y}|] \tag{34}$$

In this case, PDF already takes values in the range $[0, 1]$, so no additional transformation is required. But, to be consistent with previous stochastic indicators, function $f(\bullet)$ should also refer to the entire series $y_i^L[k]$ and not only to individual samples . To obtain this aggregated value $f_{av}(\bullet)$, we are considering the average probability (35) of all samples in the series $y_i^L[k]$.

$$f_{av}\left(y_i^L\right) = \frac{1}{L} \sum_{\forall y \in y_i^L} f(y) = \frac{1}{card\{y_i\} \bullet L} \bullet \sum_{\forall y \in y_i^L} \sum_{\forall \tilde{y} \in y_i} u[\varepsilon - |y - \tilde{y}|] \tag{35}$$

With all these three stochastics indicators, probability q may be obtained through a polynomial function (36), where h_{max}^1, h_{max}^2 and h_{max}^3 are the maximum exponents for the polynomial and coefficients σ_1, σ_2 and σ_3 are weights to control the relevance of each stochastic indicator in the calculation of probability q. To be consistent with the definition of probability, the addition of these three weights must be equal to the unit (37). These exponents h_{max}^1, h_{max}^2 and h_{max}^3 control the changing speed of probability q with the three previously described indicators. As exponents get higher, changes in entropy H_{map}, mutual information I_{map} or the PDF f_{av} cause a more significant change in probability q.

$$q = \frac{\sigma_1}{h_{max}^1} \bullet \sum_{h_1=1}^{h_{max}^1} \left(1 - H_{map}\right)^{h_1} + \frac{\sigma_2}{h_{max}^2} \bullet \sum_{h_2=1}^{h_{max}^2} \left(I_{map}\right)^{h_2} + \frac{\sigma_3}{h_{max}^3} \bullet \sum_{h_3=1}^{h_{max}^3} \left(f_{av}\right)^{h_3} \tag{36}$$

$$\sigma_1 + \sigma_2 + \sigma_3 = 1 \tag{37}$$

Finally, transactions are validated if probability q goes below a given threshold q_{th} (38). When that happens, the entire series $y_i^L[k]$ is validated.

$$ifq \leq q_{th} \Rightarrow y_i^L[k] validated \tag{38}$$

3.3 Transaction validation, block generation and consensus protocol

In the proposed security solution, edge managers $\mathcal{M}$ maintain a Blockchain ledger. Edge managers $\mathcal{M}$ receive, accumulate, and validate transactions t_j^i, describing the generation of new AmI data by sensor nodes n_i. Valid transactions t_j^i are written down in a new block b_i for the chain by edge manager m_j. Block b_i is linked to the last valid block through its hash, which will be validated by the other edge managers m_k.

In order to validate transactions, and according to the proposed data validation mechanism (see Sect. 3.2), a minimum of L different transactions from node n_i must be accumulated by edge manager m_i. But, since transactions associated to different nodes n_j can be accumulated at the same time, final blocks b_i contain T_i valid transactions t_j^i which may combine transactions referring different nodes n_j.

In our Blockchain ledger, we name $\mathcal{T}(n_i)$ the collection of all transactions that meet the conditions to be valid only considering the source node (i.e., according to the proposed reputation model, see Sect. 3.1). On the other hand, we name $\mathcal{T}(\mathcal{L})$ the set of all transactions that meet the conditions to be part of the historical record $\mathcal{L}$ of the ledger (i.e., according to the proposed data validation framework, see Sect. 3.2). Thus, valid transactions t_j^i are those contained in the intersection of both sets (39).

$$t_j^i \in \mathcal{T}(n_i) \cap \mathcal{T}(\mathcal{L}) \tag{39}$$

In that way, a block b_i is validated by consensus, if a majority of edge managers m_j validates all the individual transactions t_j^i it contains. But in the general case, the sensor nodes and edge managers are randomly distributed. Some sensor nodes can be connected to several edge managers, and the edge managers' coverage area $\mathcal{C}_i$ may contain different numbers K_i of sensor nodes $n_j^{\mathcal{C}_i}$. Then, the knowledge that each edge manager has about the AmI deployment is different. To represent this asymmetry, voting is weighted by K_i parameters. And a block b_i is validated if the aggregate knowledge K_{ag}^+ of managers supporting the block validation is superior to the aggregate knowledge K_{ag}^- of managers which do not do it (40).

$$K_{ag}^+ = \sum_{\forall m_j validates b_i} K_j > K_{ag}^- = \sum_{\forall m_j does not validate b_i} K_j \tag{40}$$

Figure 3 shows a description with details of the proposed block generation, transaction validation and consensus algorithm (executed by edge managers).

As can be seen, transactions t_j^i and blocks b_i for validation, as well as voting results v_i, arrive randomly to the edge manager. Regarding transactions t_j^i, they are stored together with all transactions $y_i[k]$ coming from the same sensor node n_i. Edge manager m_j tries to create a new block at the discrete time instants ψ_r (41). These time instants may be homogenously distributed (periodical) or can be triggered by events.

$$\{\psi_r r \in \mathbb{N}\} \tag{41}$$

At each time instant ψ_r, edge manager m_j evaluates if it stores at least L data samples (or transactions) coming from any of the nodes $n_j^{\mathcal{C}_i}$ within its coverage area $\mathcal{C}_i$. For all nodes n_i for which at least L transactions are available, the edge manager m_j validates the source node n_i through the proposed reputation framework. If node n_i is validated, the algorithm continues. If not, all associated transactions are discharged and deleted from series $y_i[k]$. For all nodes n_i whose reputation is high enough, the algorithm validates the collection of L transactions $y_i^L[k]$ through the data validation framework. If transactions are validated, the algorithm continues. If not, all transactions are discharged and deleted.

After this process, all validated transactions t_j^i are written down in a block b_i by manager m_j, which is published and distributed among all other edge managers m_k.

When an edge manager receives a new block b_i for validation, it executes the algorithm as described above: first, it validates the source node n_i using the reputation model and, later, the transaction series $y_i^L[k]$ through the data validation framework. If all transactions t_j^i within the block b_i are validated, manager m_k votes positively (and publicly) and knowledge K_j is added to aggregated knowledge K_{ag}^+. On the contrary, knowledge K_j is added to the negative aggregated knowledge K_{ag}^-. Voting results v_i are public, and all managers receive the updates. When all edge managers $\mathcal{M}$ have voted, the block b_i is definitely validated, only if the consensus is enough (40). This operation is distributed as all edge managers track the voting results.

When block b_i is definitely validated, all edge managers m_j remove from their collection of transactions to be validated $y_i^L[k]$ all transactions t_j^i included in the block b_i, as the Blockchain ledger cannot contain duplicated transactions.

4 Experimental validation

In order to evaluate the performance of the proposed security solution in the context of isolated AmI deployments, we designed and carried out an experimental validation. All experiments were supported by simulation tools and scenarios, so we can easily control variables such as the number of sensor nodes and/or edge manager within the deployment and then, analyze more deeply the behavior of the proposed technology. Section 4.1 describes the experimental methodology, while Sect. 4.2 presents and discusses the obtained results.

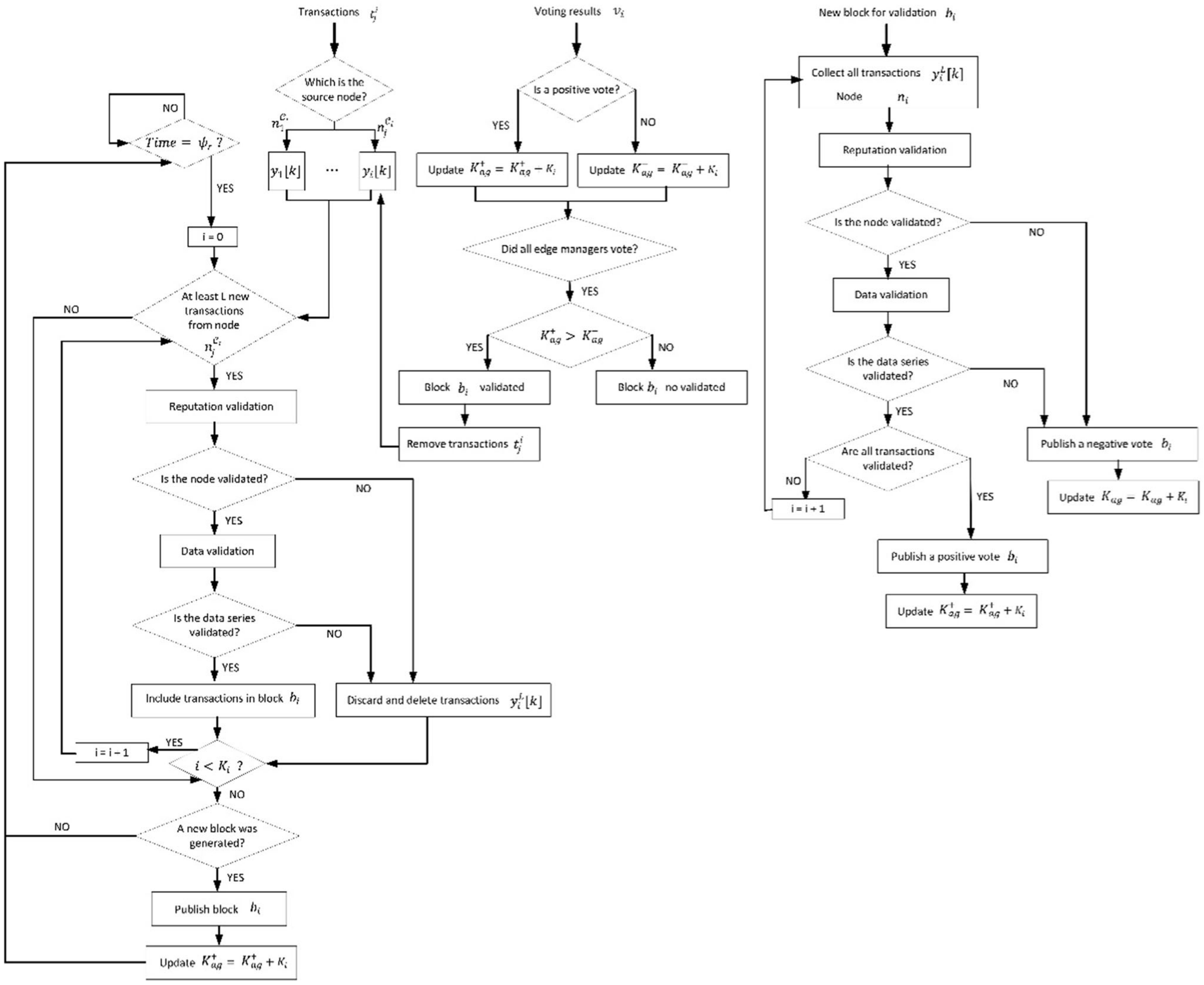

Transactions t_j^i
Which is the source node?
$n_1^{e_i}$
$n_j^{e_i}$
$y_1[k]$
…
$y_i[k]$
NO
$Time = \psi_r$?
YES
i = 0
NO
At least L new transactions from node $n_j^{e_i}$
YES
Reputation validation
Is the node validated?
NO
YES
Data validation
Is the data series validated?
NO
YES
Include transactions in block b_i
Discard and delete transactions $y_i^L[k]$
YES
i = i − 1
$i < K_i$?
NO
NO
A new block was generated?
YES
Publish block b_i
Update $K_{ag}^+ = K_{ag}^+ + K_i$
Voting results v_i
Is a positive vote?
YES
NO
Update $K_{ag}^+ = K_{ag}^+ - K_i$
Update $K_{ag}^- = K_{ag}^- + K_i$
Did all edge managers vote?
YES
$K_{ag}^+ > K_{ag}^-$
YES
NO
Block b_i validated
Block b_i no validated
Remove transactions t_j^i
New block for validation b_i
Collect all transactions $y_i^L[k]$
Node n_i
Reputation validation
Is the node validated?
NO
YES
Data validation
Is the data series validated?
NO
YES
Are all transactions validated?
NO
i = i + 1
YES
Publish a negative vote b_i
Update $K_{ag} - K_{ag} + K_i$
Publish a positive vote b_i
Update $K_{ag}^+ = K_{ag}^+ + K_i$

Fig. 3 Block generation, transaction validation and consensus algorithm

4.1 Experimental methdology, material and methods

The proposed experimental validation includes two different phases. In the first phase, the Cyber-Physical attack detection and mitigation capabilities of the proposed security solution are analyzed. Basically, the percentage of attacks successfully identified and blocked is studied. In the second phase, we focus on the performance level, and variables such as the required processing time or detection delay (critical in Blockchain-based systems) are analyzed. The objective is to identify not only whether the proposed mechanisms behave as expected, but also whether its performance is compatible with AmI deployment operations.

Specifically, five experiments were carried out. The first two experiments took place in the first experimental phase. The first experiment analyzed the percentage of Cyber-Physical attacks (false information injection) that were correctly detected and stopped, together with the number of false positive and false negative detections. The experiment was repeated for different numbers N of sensor nodes and different numbers M of edge managers in the AmI deployment. Later, the second experiment analyzed the percentage of Cyber-Physical attacks that were correctly detected and stopped too (as well as the amount of false positive and false negative detections), but in this case the experiment was repeated for two different types of attacks (fast and slow attacks). In fast attacks, false information was injected as a flood trying to collapse the AmI deployment as soon as possible. In slow attacks, false information is injected with the same speed as legitimate information, looking for a long-term impact. Different values for L parameter (employed in the data validation framework) were also considered in this second experiment.

On the other hand, the three last experiments were part of the second experimental phase. The third experiment analyze the required time (delay) to detect and mitigate a Cyber-Physical attack when using the proposed security solution, for different N of sensor nodes and different numbers M of edge managers in the AmI deployment. Additionally, the fourth experiment also studied the attack detection delay in the proposed security mechanism, but in this case for different values of L parameter (employed in the data validation framework) and two different types of attacks (fast and slow attacks, as described above). Finally, the fifth experiment studies the required processing time and computational resources (memory) required to execute the proposed security solution. Results were divided into two different groups, depending on the type of device (sensor node or edge manager).

All of these experiments were supported by a simulation scenario, built and operated using the MATLAB 2022a and Simulink software. This software was executed on a Linux-based machine (Ubuntu 22.04 LTS) with the following hardware characteristics: Dell R540 Rack 2U, 96 GB RAM, two processors Intel Xeon Silver 4114 2.2G, HD 2 TB SATA 7,2 K rpm.

In the proposed simulation model, the sensor nodes generated new data samples (or transactions) on a random basis. Edge managers were also randomly associated with sensor nodes. Sensor nodes could have three different measurement capabilities: temperature, humidity, and carbon dioxide. The proportion of each sensor type was randomly configured at every simulation. Attacks were represented by turning malicious a random percentage of sensor nodes, always below 25% of the total number available in every different simulation. Communication protocols of legitimate nodes were Bluetooth or WiFi (randomly selected), while malicious nodes could also employ LoRa protocols. Table 1 shows the proposed configuration for all parameters that are not experimental variables.

Each simulation represented seventy-two (72) hours of AmI deployment operations. All simulations were repeated twelve times to reduce the impact of exogenous effects, such as numerical errors or interferences caused by the operating system. Final results were obtained as the average of all twelve realizations.

4.2 Results

Figure 4 shows the results of the first experiment. As can be seen in Fig. 4(a), the proposed security solution is able to detect and mitigate up to 93% of Cyber-Physical attacks. Besides, even in the worst case, the detection capability is above 78%. As the Blockchain-ledger and the associated validation algorithms are distributed, the performance is significantly worse when few devices (sensor nodes and/or edge managers) are deployed within the AmI system. The best performance is obtained for deployments including between one thousand (1000) and two thousand (2000) sensor nodes. After this point, performance sightly decreases again (the success detection rate is reduced to around 2%), because of noises and numerical fluctuations caused by very large AmI deployments. But this decrease is not as relevant as the observed increase between AmI deployment with twenty-five (25) and one thousand (1000) nodes (the success detection rate increases around 19%). In conclusion, the proposed solution is especially useful for large isolated AmI deployments (more than one thousand nodes).

The impact of the number of edge managers is also observable and relevant. But this is only significant for small AmI deployments (below one thousand nodes). For this kind of deployments, systems with only five (5) edge managers only get a detection rate of 78%, while systems including one hundred (100) managers achieve an 89% rate. That is because of consensus. When few edge managers are considered, consensus is weak and block validations are not consistent (the are highly affected by numerical errors in the data validation framework, mainly). But a higher number of edge managers can mitigate this impact, thanks to a much more complex consensus. Anyway, as the AmI deployment increases in size (more sensor nodes are included), this effect disappears. Deficiencies in the consensus protocol are compensated by much abundant reputation information from the AmI deployment, as well as more samples and historical series to be employed in the data validation framework, which reduces errors and makes all configurations behave the same regardless the number of edge managers. In conclusion, for small AmI deployments, a higher number of edge managers can increase performance.

Regarding false negative detections, Fig. 4(b), they are much more common than false positive detections, Fig. 4(c). Actually, the proposed security solution is designed to specially avoid false positive detections that perturbed the AmI deployment operations, since false information (in small amounts) is usually risk-free. The false positive detection rate decreases monotonously as the success detection rate increases, and for large AmI deployments keep under 0.3% in all cases. The false negative detection rate behaves similarly, but in this case the minimum value is around 3%. In addition, the rate also increases sightly for very large AmI deployments, which is consistent with the reduction in the success detection rate.

Figure 5 shows the results of the second experiment. This experiment was carried out for an AmI deployment with one thousand (1000) sensor nodes and twenty-five (25) edge managers. As can be seen, the evolution for all

Table 1 System configuration

Parameter	Value	Comments	Parameter	Value	Comments
ψ_r	10 s	Blocks are generated periodically	$\sigma_1, \sigma_2, \sigma_3$	$\frac{1}{3}$	All stochastic parameters are equally relevant
q_{th}	0.2	False data probability below 20%	π	4	Four discrete time units tolerance
ε	0.15	15% tolerance	R_{th}	0.5	Reputation threshold 50%
$h^1_{max}, h^2_{max}, h^3_{max}$	4	Forth order polynomial function	r, Q	2	
λ_j	$\frac{1}{N}$	All nodes are equally important	$\Sigma^{pos}_{max}, \Sigma^{neg}_{max}$	25	
K_{av}	0.15		cl	1	All capabilities are equally critical

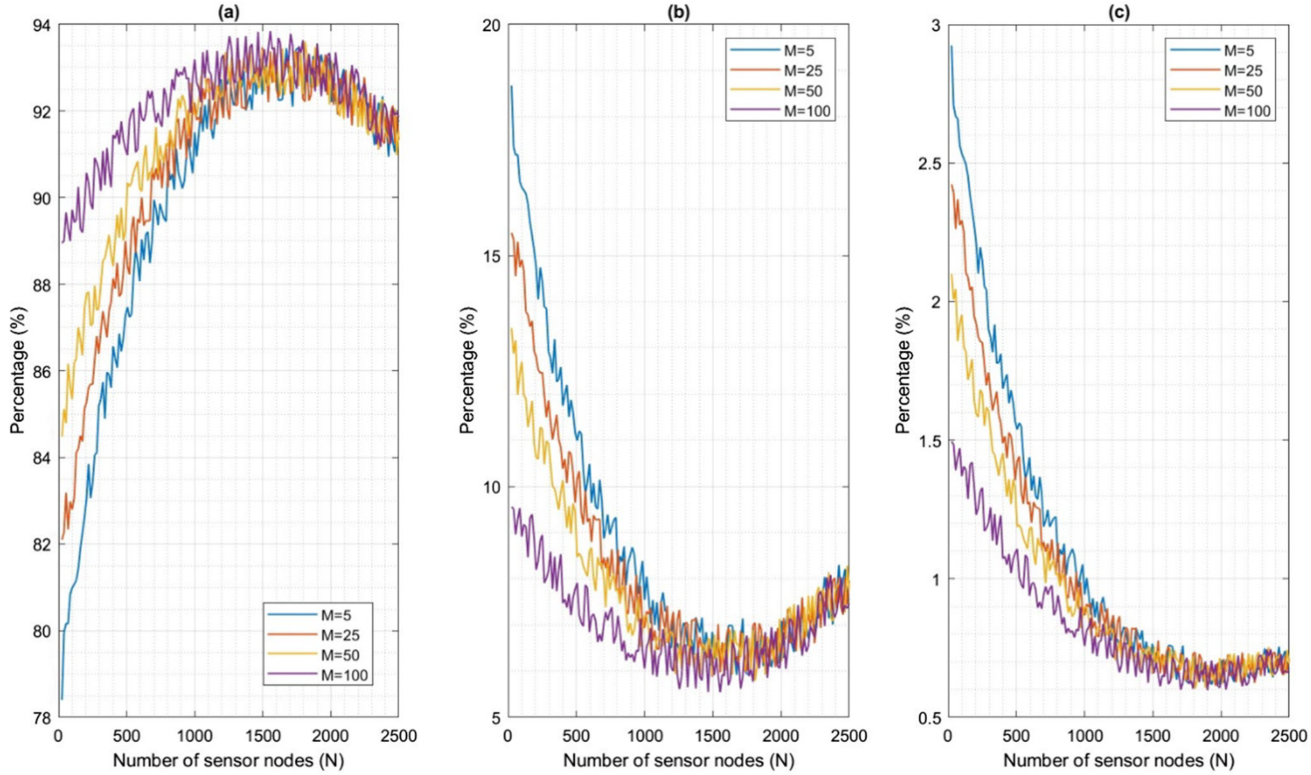


Fig. 4 Results for the first experiment. **a** True positive detections. **b** False negative detections. **c** False positive detections

rates is qualitatively similar to the results in Fig. 4. Thus, both experiments are coherent.

For low values of L parameter fast attacks are more efficiently detected. In fast attacks, the impact is more relevant, even in short time periods, so the data validation framework may detect malicious behavior even with a limited number of samples (or transactions). But if the value for L parameter gets even smaller, performance get worse in any situation, with a success detection rate around 60% for fast attacks and around 40% for slow attacks. As can be seen in Fig. 5(a) the optimal point for fast attacks is $L \approx 75$, when the success detection rate is around 93%. When L parameter goes beyond this point, too many transactions must be accumulated, and fast attacks may finish and complete their objective before they are detected. On the contrary, slow attacks present a very bad behavior for low and medium values of L parameter, but it gets better monotonously as L parameter increases. For values $L > 100$ the success detection rate goes above 90%, with a maximum rate of 91%. Taking into account this analysis, a good balanced configuration could be $L = 100$. For this value, both types of attack are successfully detected with rates greater than 90%.

Regarding false-positive and false-negative detection rates, we can distinguish two regions. For low values of L parameter, false positive detection and false negative detection rates present similar values. Mainly because for such low values, the data validation framework is unstable and numerical errors make false positive detections and false negative detections occur randomly. But, for high values of L parameter, the false positive detection rate is low for both kinds of attacks (fast and slow). Although for fast attacks the rate increases sightly (minimum value is around 2.5%, and grows up to 8%, approximately), and for slow attacks the rate decreases monotonously (minimum rate, 2.5%). On the contrary, false negative detection rate for fast attacks greatly increases its value. Increasing from 3% (minimum value for $L = 75$) to 19%. Meanwhile, for slow attacks, this rate also decreases monotonously and stays around 5%.

On the other hand, Fig. 6 shows the results of the third experiment (second experimental phase). In this experiment, only attacks successfully detected were considered. As can be seen, attack detection delays evolve linearly with the number of sensor nodes within the AmI deployment, as well as with the number of edge managers. This is consistent with the proposed validation and consensus algorithm, as the voting process requires all edge managers to vote (so it is a longer process as more managers participate). Furthermore, since more nodes are included in the AmI deployment, the blocks tend to include more transactions, which also increases (because of the data validation framework and the reputation model) the detection delay (linearly, see Fig. 3). If we analyze the situation where the highest success detection rate was reported (one thousand nodes), the delay is between eleven (11) and thirty-seven (37) seconds. These values are acceptable for most AmI deployments and Cyber-Physical attacks, as they are resilient enough to handle and attack for such a short time.

Figure 7 shows the results for the third experiment. For the second experiment, these results represent an AmI deployment with one thousand (1000) devices. And only successful attack detections were considered. In this case, the differences between slow and fast attacks are very reduced. Delays are only 7% higher for slow attacks because, on some occasions, attacks are so slow that more than one block is needed to detect it, contrary to fast attacks. Time also evolves linearly with parameter L, as stochastic indicators in the data validation framework are obtained using sequential loops over all accumulated samples. If we consider a balanced value for L parameter, for example $L = 75$, detection delay is around 37.5 s for slow attacks and 29 s for fast attacks. As said above, these values are acceptable for most AmI deployments, so we can conclude that the proposed solution is adequate for isolated AmI deployments.

Finally, Table 2 shows the results of the fifth experiment. This analysis was carried out for the configuration with the best behavior (one thousand sensor nodes, twenty-

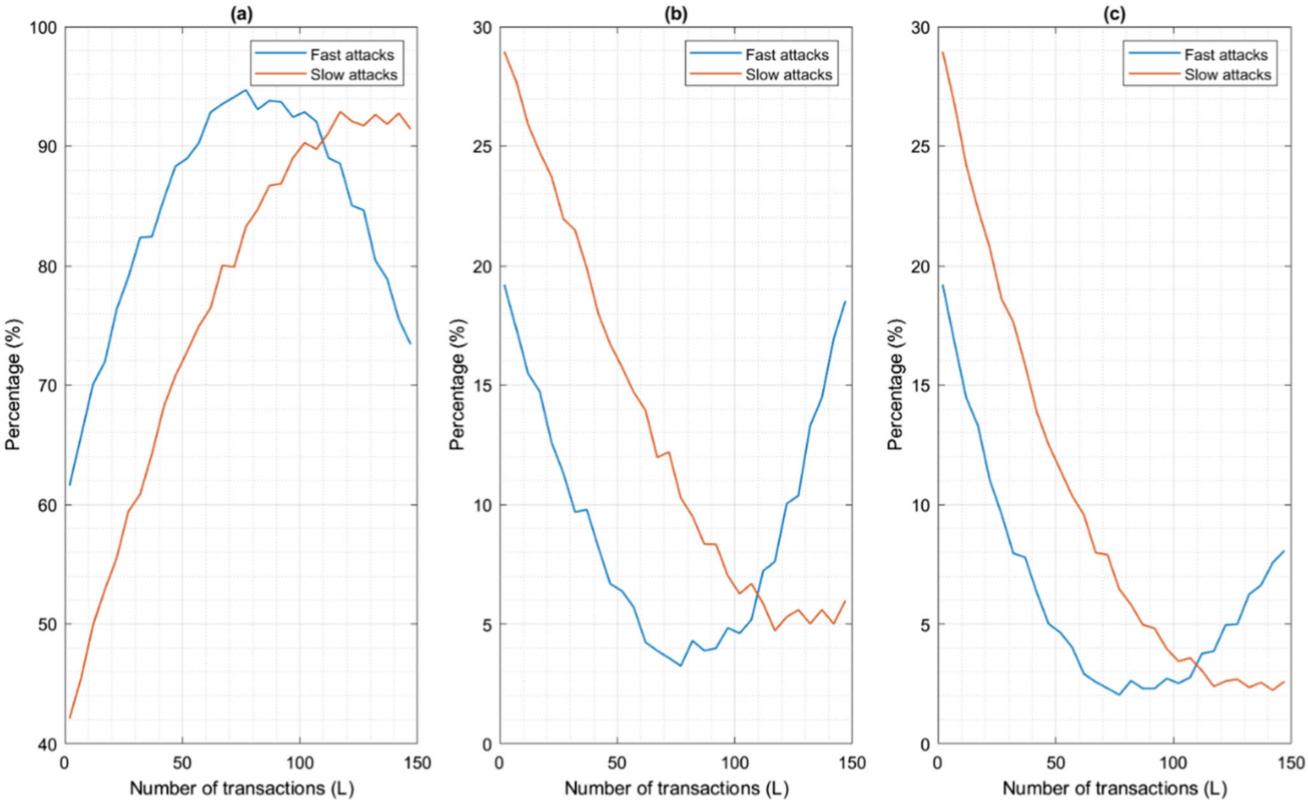


Fig. 5 Results for the second experiment. **a** True positive detections. **b** False negative detections. **c** False positive detections

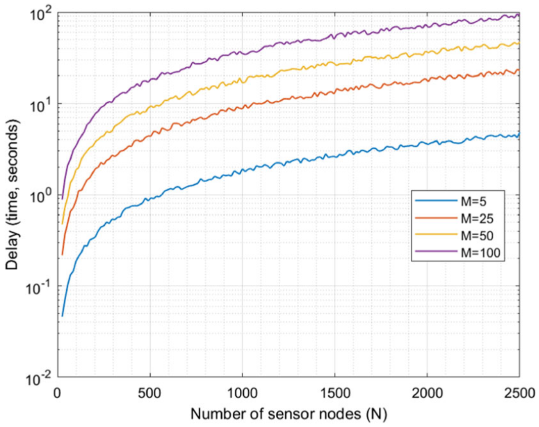


Fig. 6 Results for the third experiment. Detection delay

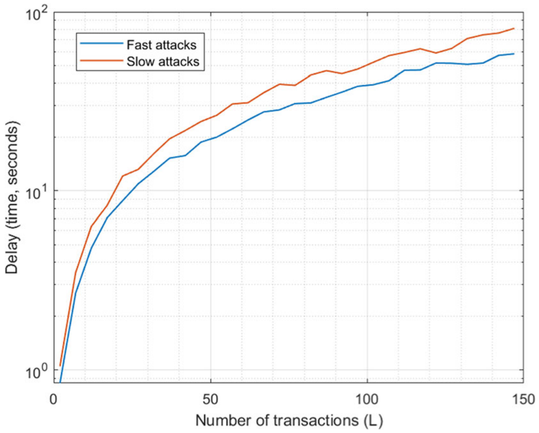


Fig. 7 Results for the fourth experiment. Detection delay

Table 2 Results for the fifth experiment. Resource consumption

Device	Processing delay	RAM memory	Code memory
Sensor node	119 ms	11%	9.3%
Edge manager	41.1 s	12%	14.4%

five edge manager and $L = 100$). As can be seen, around 10% of the resources in the sensor nodes are consumed by the proposed solution. To do these calculations, the ESP32 microcontroller was taken as a reference. Processing delay, in addition, is acceptable for most applications, where sensor nodes generate data every few seconds. On the other hand, resource consumption in edge managers is slightly higher. Around 15% memory is required by the proposed Blockchain-based solution, and most part of the required attack detection delay is assumed by edge managers (on average more than forty seconds). To obtain these results, an Artik 530 architecture was considered. In conclusion, the resource consumption of the proposed security solution is compatible with the characteristics of AmI deployments.

5 Conclusions

In this paper, we propose a new decentralized security solution, based on a Blockchain ledger, to protect isolated Ambient Intelligence deployments. In this ledger, new sensing data are considered transactions that must be validated by edge managers, which operate a Blockchain network. This validation is based on reputation metrics evaluated by sensor nodes using historical network data and identity parameters. Through information theory, the coherence of all transactions with the behavior of the historical deployment is also analyzed and considered in the validation algorithm. The relevance of edge managers in the Blockchain network is also weighted considering the knowledge they have about the deployment.

Five different experiments are provided, showing that the attack detection rate can achieve values of up to 93%, with a detection delay of 37 s in the worst case. Additionally, resource consumption in sensor nodes and edge managers in AmI deployments is acceptable for most current hardware platforms. As a result, we can conclude that the proposed security solution is adequate for isolated AmI deployments.

For future work, the proposed solution will be validated in a real scenario, considering hardware sensor nodes and edge manager within a functional isolated AmI deployment.

Acknowledgements This work is supported by Comunidad de Madrid within the framework of the Multiannual Agreement with Universidad Politécnica de Madrid to encourage research by young doctors (PRINCE project).

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Data availability Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Bordel, B., Alcarria, R., Robles, T., & Martín, D. (2017). Cyber–physical systems: Extending pervasive sensing from control theory to the internet of things. *Pervasive and mobile computing, 40*, 156–184.
2. Gams, M., Gu, I. Y. H., Härmä, A., Muñoz, A., & Tam, V. (2019). Artificial intelligence and ambient intelligence. *Journal of Ambient Intelligence and Smart Environments, 11*(1), 71–86.
3. Bordel, B., Alcarria, R., & Robles, T. (2022). Recognizing human activities in Industry 4.0 scenarios through an analysis-modeling-recognition algorithm and context labels. *Integrated Computer-Aided Engineering, 29*(1), 83–103.
4. Tange, K., De Donno, M., Fafoutis, X., & Dragoni, N. (2020). A systematic survey of industrial internet of things security: Requirements and fog computing opportunities. *IEEE Communications Surveys & Tutorials, 22*(4), 2489–2520.
5. Bordel, B., Alcarria, R., Sánchez-de-Rivera, D., & Robles, T. (2017). Protecting industry 4.0 systems against the malicious effects of cyber-physical attacks. In Ubiquitous Computing and Ambient Intelligence: 11th International Conference, UCAmI 2017, Philadelphia, PA, USA, November 7–10, 2017, Proceedings (pp. 161–171). Springer International Publishing.
6. Stankovic, J. A., Ma, M., Preum, S. M., & Alemzadeh, H. (2021). Challenges and directions for ambient intelligence: A cyber physical systems perspective. In 2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI) (pp. 232–241). IEEE.
7. Robles, T., Bordel, B., Alcarria, R., & de Andrés, D. M. (2017). Mobile wireless sensor networks: Modeling and analysis of three-dimensional scenarios and neighbor discovery in mobile data collection. *Ad Hoc Sens. Wirel. Networks, 35*(1–2), 67–104.
8. Bordel, B., Alcarria, R., Sanchez de Rivera, D., Martín, D., & Robles, T. (2018). Fast self-configuration in service-oriented smart environments for real-time applications. *Journal of Ambient Intelligence and Smart Environments, 10*(2), 143–167.
9. Schmeelk, S., Roth, S., Rooney, J., Tariq, M., Wood, K., Kamen, J., & Dragos, D. (2022). Ambient intelligence security checks: Identifying integrity vulnerabilities in industry scripts. In Intelligent Systems and Applications: Proceedings of the 2022 Intelligent Systems Conference (IntelliSys) Vol. 3 (pp. 590–599). Cham: Springer International Publishing.
10. Bordel, B., Alcarria, R., Robles, T., & Sánchez-Picot, Á. (2018). Stochastic and information theory techniques to reduce large datasets and detect cyberattacks in ambient Intelligence environments. *IEEE Access, 6*, 34896–34910.
11. Dunne, R., Morris, T., & Harper, S. (2021). A survey of ambient intelligence. *ACM Computing Surveys (CSUR), 54*(4), 1–27.
12. Kim, J. C., & Chung, K. (2020). Neural-network based adaptive context prediction model for ambient intelligence. *Journal of Ambient Intelligence and Humanized Computing, 11*, 1451–1458.
13. Cao, K., Liu, Y., Meng, G., & Sun, Q. (2020). An overview on edge computing research. *IEEE Access, 8*, 85714–85728.
14. Rodríguez-Pérez, N., Toledo-Castro, J., Caballero-Gil, P., Santos-González, I., & Hernández-Goya, C. (2022). Secure ambient intelligence prototype for airports. *Journal of Ambient Intelligence and Humanized Computing*, *13*, 5405–5417.
15. Qazi, R., Qureshi, K. N., Bashir, F., Islam, N. U., Iqbal, S., & Arshad, A. (2021). Security protocol using elliptic curve cryptography algorithm for wireless sensor networks. *Journal of Ambient Intelligence and Humanized Computing, 12*, 547–566.
16. Suratkar, S., Shah, K., Sood, A., Loya, A., Bisure, D., Patil, U., & Kazi, F. (2022). An adaptive honeypot using Q-Learning with severity analyzer. *Journal of Ambient Intelligence and Humanized Computing, 13*(10), 4865–4876.
17. Zhang, L., Feng, G., Qin, S., Sun, Y., & Cao, B. (2022). Access control for ambient backscatter enhanced wireless internet of things. *IEEE Transactions on Wireless Communications, 21*(7), 5614–5628.
18. Lee, O. J., Nguyen, H. L., Jung, J. E., Um, T. W., & Lee, H. W. (2017). Towards ontological approach on trust-aware ambient services. *IEEE Access, 5*, 1589–1599.
19. Rathee, G., Kerrache, C. A., & Calafate, C. T. (2022). An Ambient Intelligence approach to provide secure and trusted Pub/Sub messaging systems in IoT environments. *Computer Networks, 218*, 109401.
20. Saini, N. K. (2016). Trust factor and reliability-over-a-period-of-time as key differentiators in IoT enabled services. In 2016 International Conference on Internet of Things and Applications (IOTA) (pp. 411–414). IEEE.
21. Nguyen, H. L., Lee, O. J., Jung, J. E., Park, J., Um, T. W., & Lee, H. W. (2017). Event-driven trust refreshment on ambient services. *IEEE Access, 5*, 4664–4670.
22. Quadar, N., Chehri, A., Jeon, G., Hassan, M. M., & Fortino, G. (2022). Cybersecurity issues of IoT in ambient intelligence (AmI) environment. *IEEE Internet of Things Magazine, 5*(3), 140–145.
23. El-Dosuky, M. A., & Eladl, G. H. (2019). SPAINChain: security, privacy, and ambient intelligence in negotiation between IoT and Blockchain. In New Knowledge in Information Systems and Technologies: Vol. 1 (pp. 415–425). Springer International Publishing.
24. Florea, A. I., Anghel, I., & Cioara, T. (2022). A review of Blockchain technology applications in ambient assisted living. *Future Internet, 14*(5), 150.
25. Rathee, G., Balasaraswathi, M., Chandran, K. P., Gupta, S. D., & Boopathi, C. S. (2021). A secure IoT sensors communication in industry 4.0 using Blockchain technology. *Journal of Ambient Intelligence and Humanized Computing, 12*, 533–545.
26. Mkpa, A., Chin, J., & Winckles, A. (2019). Holistic Blockchain approach to foster trust, privacy and security in IoT based ambient assisted living environment. In 2019 15th International Conference on Intelligent Environments (IE) (pp. 52–55). IEEE.
27. Bordonaro, A., De Paola, A., Re, G. L., & Morana, M. (2020). Smart auctions for autonomic ambient intelligence systems. In 2020 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 180–187). IEEE.
28. Alam, T., Ullah, A., & Benaida, M. (2022). Deep reinforcement learning approach for computation offloading in Blockchain-enabled communications systems. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-021-03663-2
29. Buhalis, D. (2020). Technology in tourism-from information communication technologies to eTourism and smart tourism towards ambient intelligence tourism: A perspective article. *Tourism Review, 75*(1), 267–272.
30. Makhdoom, I., Tofigh, F., Zhou, I., Abolhasan, M., & Lipman, J. (2020). PLEDGE: A proof-of-honesty based consensus protocol for blockchain-based IoT systems. In 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (pp. 1–3). IEEE.
31. Alrubei, S., Ball, E., & Rigelsford, J. (2021). Securing IoT-blockchain applications through honesty-based distributed proof of authority consensus algorithm. In 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA) (pp. 1–7). IEEE.

32. Makhdoom, I., Tofigh, F., Zhou, I., Abolhasan, M., & Lipman, J. (2020). PLEDGE: An IoT-oriented Proof-of-Honesty based Blockchain Consensus Protocol. In 2020 IEEE 45th Conference on Local Computer Networks (LCN) (pp. 54–64). IEEE.
33. Bordel, B., & Alcarria, R. (2022). Distributed trust and reputation services in pervasive internet-of-things deployments. In Mobile Internet Security: 5th International Symposium, MobiSec 2021, Jeju Island, South Korea, October 7–9, 2021, Revised Selected Papers (pp. 16–29). Singapore: Springer Nature Singapore.
34. Bordel, B., Alcarria, R., De Andrés, D. M., & You, I. (2018). Securing internet-of-things systems through implicit and explicit reputation models. *IEEE Access, 6*, 47472–47488.
35. Sun, H., Tan, Y. A., Zhu, L., Zhang, Q., Ai, S., & Zheng, J. (2022). A blockchain-based access control protocol for secure resource sharing with mobile edge-cloud collaboration. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-022-04020-7
36. Heshmati, A., Bayat, M., Doostari, M., & Pournaghi, S. M. (2023). Blockchain based authentication and access verfication scheme in smart home. *Journal of Ambient Intelligence and Humanized Computing, 14*(3), 2525–2547.
37. Ponce, V., & Abdulrazak, B. (2022). Ambient intelligence governance review: From service-oriented to self-service. *PeerJ Computer Science, 8*, e788.
38. Wilkowska, W., Offermann, J., Spinsante, S., Poli, A., & Ziefle, M. (2022). Analyzing technology acceptance and perception of privacy in ambient assisted living for using sensor-based technologies. *PLoS ONE, 17*(7), e0269642.
39. Vourganas, I., Attar, H., & Michala, A. L. (2022). Accountable, responsible, transparent artificial intelligence in ambient intelligence systems for healthcare. In Intelligent Healthcare: Infrastructure, Algorithms and Management (pp. 87–111). Singapore: Springer Nature Singapore.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Borja Bordel received the B.S. and M.S. degrees in telecommunication engineering from the Technical University of Madrid, in 2012 and 2014, respectively, and the Ph.D. degree in 2018. He is currently an Associate Professor with the Computer Science School. His research interests include cyber-physical systems, wireless sensor networks, radio access technologies, communication protocols, and complex systems. https://orcid.org/0000-0001-7815-5924.

Ramón Alcarria received the M.S. and Ph.D. degrees in telecommunication engineering from the Technical University of Madrid, in 2008 and 2013, respectively. He is currently an Associate Professor with the Department of Geospatial Engineering, Technical University of Madrid. He has been involved in several Research and Development European and National projects related to Future Internet, the Internet of Things, and Service Composition. His research interests include service architectures, sensor networks, human–computer interaction, and prosumer environments. E-mail: ramon.alcarria@upm.es, https://orcid.org/0000-0002-1183-9579.

Tomás Robles is a full professor at the E.T.S.I Telecommunication of the Technical University of Madrid, UPM. He received a M.S. and Ph.D. degrees in Telecommunication Engineering from Technical University of Madrid in 1987 and 1991 respectively. His research interests are advanced applications and services for broadband networks and technology for engineering education. E-mail: tomas.robles@upm.es, https://orcid.org/0000-0002-6940-8421.