Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

A Trust Assurance Technique for IoT based on Human Behavior Compliance

Marco Anisetti*[1] | Claudio Agostino Ardagna[1] | Ernesto Damiani[2,1] | Alessandro Sala[3]

[1]Universitá degli Studi di Milano, Dipartimento di Informatica, Milano, Italy
[2]Khalifa University, Centre on Cyber-Physical Systems, Abu Dhabi, UAE
[3]HUKO, Nextdev srl, Chiari, Italy

Correspondence
*Marco Anisetti, Email: marco.anisetti@unimi.it

Summary

The advent of the Internet of Things (IoT) has radically changed the way in which computations and communications are carried out. People are just becoming another component of IoT environments and, in turn, IoT environments are becoming a mixture of platforms, software, services, things, and people. The price we pay for such dynamic and powerful environment is an intrinsic uncertainty and low trustworthiness due to its opaque perimeter, the multitude of different data sources with unknown providers, and uncertain responsibilities. Trustworthiness of observables collected by smart devices (from minuscle sensors to bigger machines) is fundamental to build a chain of trust on a decision process taken according to these observables. Some assurance solutions evaluate the quality of collected data, though they are difficult to apply in IoT environments for performance and cost reasons. In this paper, we take a different approach and put forward the idea that, in many cases, the behavior of people owning smart devices can contribute to the evaluation of the trustworthiness of collected data and, in turn, of the whole decision process. We therefore define an assurance methodology based on data analytics evaluating the compliance of people to behavioral policies. The more people behavior is compliant, the higher the trustworthiness of data collected through their smart devices.

KEYWORDS:
Assurance, Behavioral Analysis, Big Data, Internet of Things

1 | INTRODUCTION

The success of the Internet of Things (IoT) is increasingly pushing towards the development of Cyber-Physical Systems (CPSs) that go beyond traditional IT boundaries, combining physical and digital environments in a single one. Today systems have an opaque perimeter, where a mixture of platforms, software, services, things, and people collaborate in an opportunistic way. Computations are then moving from the center (e.g., cloud) to the periphery, supporting analytics and knowledge extraction partially executed at the edge of the network, near the physical environment and sensors where data are collected. In this scenario, people carry/manage a plethora of smart devices (often integrated in their smartphones or in their homes) sensing the surrounding environment and communicating with edge nodes without even noticing. Like for the Web 2.0 revolution when user-generated content entered the loop, people are not only passive consumers of pervasive services; they rather become (often unintentional) service providers that distribute user-collected data. The exponential growth of smart devices (200billions of connected objects with a mean of 26 objects for every human predicted by 2020[1]) and connected people (2.87billion users carrying a smartphone

by 2020[2]), coupled with pervasive mobility and the proliferation of IoT applications, make the trustworthiness of collected data the most important requirement to the success of these applications. Data trustworthiness is mandatory to build a chain of trust on a decision process taken according to IoT data and edge computations. This is very similar to the Web 2.0 scenario, where the plague of fake news and fake data substantially decreased the quality of decision processes, such as, for instance, in case of Twitter bots able to support or defame a specific product with high rates of success[3,4].

Traditional solutions to data validation mostly focused on assurance or reputation techniques. On one side, the system generating data undergoes an assurance process based on testing or monitoring[5]; on the other side, the reputation of the entity owning the system is evaluated. Both approaches are however not viable in complex cyber-physical scenarios based on smart devices for different reasons: *i)* devices are usually resource constrained and cannot therefore be the target of extensive testing/monitoring; *ii)* IoT systems have fuzzy perimeters that are difficult to evaluate using traditional assurance techniques; *iii)* devices are often owned by unknown users, whose reputation cannot be easily evaluated; *iv)* people are often unintentional data providers, differently from Human-Provided Services[6,7] where users consciously distribute services. Other approaches have provided solutions based on behavioral analysis, aimed to understand and differentiate human behaviors[8,9,10].

In this paper, we aim to fill in the above gaps by providing a new human-centric approach, where the behavior of people carrying/managing smart devices is an indicator of the trustworthiness of the generated data and corresponding devices. We define an assurance methodology based on data analytics that selects trustworthy devices according to the behavior of involved people. First, a technology-independent *assurance policy template* specifies requirements on correct human behaviors and data collection processes. Second, the assurance policy template is translated in a technology-dependent *assurance policy instance* specifying corresponding activities for human-behavioral modeling and data collection. Finally, trustworthy smart devices (and corresponding data) are selected according to a specific evaluation function and used to build a chain of trust underpinning a specific decision/inference.

The contribution of this paper is twofold: *i)* it clarifies the role of human behavior in guaranteeing the trustworthiness of IoT environments; *ii)* it defines an assurance methodology based on policies to evaluate the compliance of human behavior and, in turn, the trustworthiness of collected data and corresponding smart devices.

The remaining of the paper is organized as follows. Section 2 presents an overview of the proposed methodology and application scenario. Section 3 defines the assurance policy at the core of our methodology and describes its evaluation process. Section 4 defines the composite assurance policy for modeling complex scenarios and its evaluation process. Section 5 introduces the concept of policy portability. Section 6 presents our experimental evaluation in a real-world scenario involving smartphone traces of people moving in Chinese malls controlled by Huko[†] pervasive location system. Section 7 presents the related work and Section 8 gives our concluding remarks.

2 | APPLICATION SCENARIO AND ASSURANCE METHODOLOGY

Our reference model is a cyber-physical system where smart devices sense the surrounding environment and collect data at the basis of a specific application execution or decision process. Collected data can be pre-computed and aggregated at the edge of the system (e.g., for privacy reasons), and then forwarded to the center for the final analysis. The participating entities are: *i)* people carrying or managing smart devices; *ii)* smart devices collecting data supporting the execution of a specific application or process; *iii)* edge nodes pre-analyzing (e.g., anonymizing) data collected by smart devices; *iv)* core computing infrastructure deploying the process/application that use data computed by edge nodes; it is connected to the periphery through the edge network. Our goal is to implement an assurance process that evaluates the trustworthiness of the collected data in relation with the implemented process/application.

2.1 | Application Scenario

Our application scenario considers the problem of increasing the selling effectiveness of stores in a *Shopping Mall*, by measuring the quality of relationships between the mall and shops. This can be calculated by Key Performance Indicators (KPIs), like foot traffic counts, proximity traffic, capture rate, which are computed by tracking customers' movements within the mall (e.g., where and for how long customers stay in the mall/specific shops). To this aim, we assume that the subset of customers carrying

[†]Huko is a TM of NextDev srl, an italian software company providing solutions for IoT based-applications.

a mobile device with WiFi activated is a good representative of all customers visiting the mall and consider a set of ad hoc WiFi access points deployed on the field. Access points sense the presence of a mobile device and transfer the observed device Received Signal Strength Indicator (RSSI), along with its MAC address, to a central point of computation. RSSI is then used to locate the mobile device and its proximity (near to the entrance or inside) to a specific shop in the mall. We note that customers are unaware that they are tracked for computing KPIs of interest for the mall. Privacy issues introduced by customer tracking, even if strongly relevant, is not considered in this application scenario and will be treated in our future work. We note, however, that an anonymization process might be executed at the edge to protect users' privacy before data are transferred to the central point of computation.

Example 2.1. Let us consider the RSSI received by a mobile device of a person in the shopping mall application scenario. RSSI is affected by obstacles between the emitting and the receiving antennas. Human behavior can be considered as an obstacle and has an impact on the sensor signal, that is, RSSI varies if the mobile device is in the pocket, bag or hands of a person. For instance, when the mobile device is in the pocket, it might appear far from the source antenna, while it is very near. The level of impact depends on the scope of the computation, from *no impact* when RSSI is just used to count connected devices to *high impact* when RSSI is used to position the device.

2.2 | Assurance Methodology

Traditional assurance solutions do not consider human behavior in the evaluation of data trustworthiness and, in turn, of the validity of the corresponding processes/applications. They rather implement a verification process where either the system or devices generating data are tested/monitored[5,11,12], or the reputation of the entity owning the system or devices is evaluated. Both these approaches are however not viable in complex IT scenarios based on a multitude of smart devices for different reasons, which can be summarized as follows.

- **Technological challenge:** current systems are composed of a huge amount of heterogeneous and dynamic smart devices. Such devices are usually resource constrained, suffer from battery limitations, are often owned by unknown users, and need to adapt to a surrounding environment that is continuously changing. Assurance evaluation must depart from traditional solutions testing and monitoring each object with an IP address for technical reasons. On one side, smart devices cannot undergo complex verification process that would impair their functionalities; on the other side, the assurance infrastructure falls short of managing heterogeneous devices, each with specific requirements and peculiarities.
- **Geographical challenge:** current systems have fuzzy perimeters and unknown topologies that suggest the need of rethinking traditional assurance techniques. They can involve nomadic actors whose position is continuously changing, as well as a dynamic environment evolving over time. This environment makes assurance evaluation harder than before, and impedes the assurance evaluation of the whole system and its machines.
- **Societal challenge:** lack of consciousness is one the main problems affecting current systems. Differently from Human-Provided Services[6,7] where users consciously provide services, people are increasingly becoming unintentional data/service providers. As a consequence, they do not always follow those good practices that are mandatory for the correct execution of a given service. In this scenario involving thousands of small objects, reputation cannot be easily evaluated as well, because data providers are often unknown.
- **Legal challenge:** data management comes with legal issues that need to be properly addressed. It must cope with scenarios where ownerships and responsibilities are blurred. In fact, differently from traditional IT where perimeters and responsibilities are clear, cyber-physical systems involve smart devices often found in private households and public places, operated by unknown people. Proper data management is difficult to achieve and it is unlikely that owners have the knowledge, resources, and motivation for supporting it.

To address the above challenges, we propose a novel methodology (depicted in Figure 1) that puts people behavior (e.g., mobility, configuration) at the core of an assurance evaluation process having the following peculiarities

- **Lightweight:** it reduces the need to test/monitor a device and provides a complete assurance evaluation based on global data collected by CPS. [*Technological Challenge*]

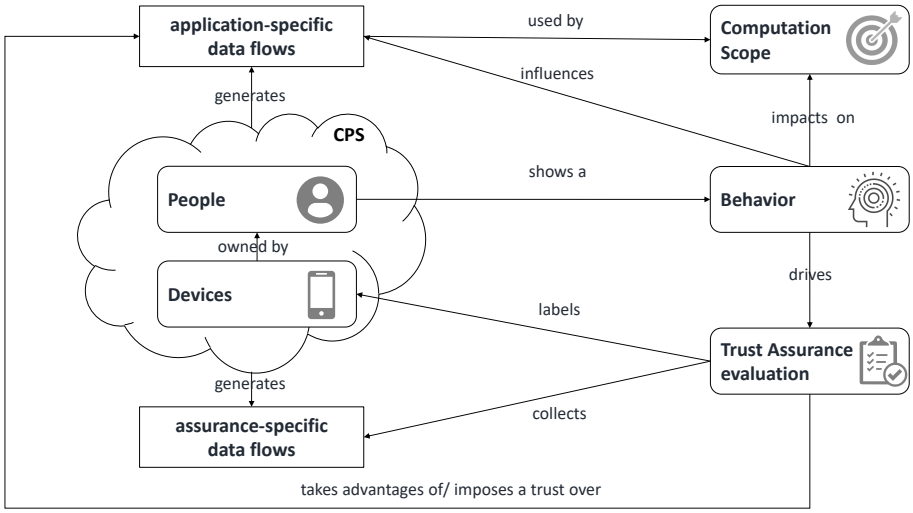

FIGURE 1 An abstract view of our methodology.

- **Model-driven:** it models the relation between application goals changing over time and assurance objectives, to support fast adaptation to new goals, on one side, and to new devices and collection mechanisms, on the other side. [*Technical Challenge*]
- **Incremental:** it copes with partial and non-uniform data flows coming at different rates, to produce or evaluate incremental models. [*Geographical, Societal Challenges*]
- **Policy-driven:** it captures relations between devices and computational objectives using a policy-based approach. [*Societal, Legal Challenges*]
- **Data-oriented:** it implements a Big data computation to evaluate the behavior of a cyber-physical system. [*Technical, Geographical, Societal, Legal Challenges*]

We put forward the idea that having a trustworthy indicator of the human behavior permits to improve the reliability of the cyber-physical system (shopping mall in our scenario) in relation to a specific goal or computation. We then depart from a scenario where people are sources of information, that is, generate data based on their own computation capabilities [6,7]. We rather consider people as another type of smart devices whose "configuration and behavior" impact on the working of surrounding devices.[‡] Figure 1 shows a CPS scenario where *people* own *devices* and are immersed into a pervasive *CPS*. Such system generates *application-specific data flows* used for the scope of the computation (e.g., a KPI-based metric computation), as well as *assurance-specific data flows* are not related to the application and can be used for behavioral assurance. Each person in the CPS under evaluation shows a *behavior* that may impact application-specific data flows and thus indirectly impacts on the computation, possibly injecting untrusted application-specific data flows. This behavior drives the *trust assurance evaluation* process, which collects specific behavioral data from the CPS or uses application-specific data flows to compute the level of trustworthiness of a given person owning specific devices.

3 | TRUST ASSURANCE EVALUATION

Our methodology evaluates the trustworthiness of smart devices and their data by means of a trust assurance evaluation of the human behavior. It implements a trust assurance evaluation process that specifies the steps to follow to retrieve evidence of the

[‡]Similar to the concept of service container in Web Services, where the behavior of the container impacts the contained services.

people behavior and the function to map this behavior to trust values. It defines requirements on both physical behavior (e.g., mobility) and digital behavior (e.g., device configuration), impacting the smart device trustworthiness in a tangible way. More in details our process is composed of three steps discussed in the following of this section. The first step consists of the definition of an *assurance policy template* (Section 3.1), which models, at an abstract level, all activities to be done for the evaluation of data and smart device trustworthiness, namely: *i)* how to collect evidence from the CPS, *ii)* how to use collected evidence to generate a behavioral model targeting the human behavior of interest, and *iii)* how to use these behaviors to label each device with a trust value. The second step employs a semi-automatic process that translates the assurance policy template in a ready-to-be-executed *assurance policy instance* (Section 3.2), where *i)* the feature collection process and the behavioral model are fully specified and *ii)* the evaluation function for trust value labeling is specified. The third step puts together all artifacts in the policy instance to evaluate human behavior and, in turn, the trustworthiness of corresponding devices and collected data (Section 3.3).

3.1 | Assurance Policy Template

Assurance policy template $\mathcal{P}^t$ defines an abstract specification of the evaluation process that represents the basis for verifying the user behavior according to predefined requirements. It is formally defined as follows.

Definition 1 (Assurance Policy Template $\mathcal{P}^t$)**.** An assurance policy template $\mathcal{P}^t$ models a technology-independent process as a triple $<\mathcal{B},\mathcal{F},\mathcal{E}>$ supporting the assurance evaluation of human behaviors, where

- $\mathcal{B}$ specifies a technology-independent *Big Data computation* used to generate the behavioral model $\mathcal{M}_B$ for the CPS under evaluation based on a set of features f and on a set of humans behaviors $b = [b_1 \cdots b_n]$ of interest.
- $\mathcal{F}$ describes the *feature collection model* according to the CPS under evaluation and features f in $\mathcal{B}$.
- $\mathcal{E}$ defines a technology-independent *Big Data computation* that specifies how to annotate a smart device with a behavior b_i according to behavioral model $\mathcal{M}_B$ in $\mathcal{B}$.

More in detail, $\mathcal{B}$ composes abstract services in a Big Data computation to produce the behavioral model $\mathcal{M}_B$ of interest. The owner of the whole process can define its own configurations, including specific features of interest to be ingested in the Big Data process, the expected behaviors, and the like. We note that the definition of the Big Data computation in $\mathcal{B}$ requires a substantial amount of data science skills and can become a problem for many users. Approaches such as the one in [13] providing a Model-based Big Data Analytics-as-a-Service (MBDAaaS) methodology, where users specify declarative requirements and smarter engine are employed to semi-automatically carry out the computation according to the specified requirements, can support users in this activity. Following MBDAaaS methodology [13], we formally define $\mathcal{B}$ as follows.

Definition 2 ($\mathcal{B}$)**.** Big Data computation $\mathcal{B}$ is implemented as a technology-independent composite process, where abstract component services are composed to model a specific Big Data pipeline, including common services for data preparation, analytics, and processing. It can also specify service configurations and features f of interest, as well as the target human behaviors b.

We note that the definition of $\mathcal{B}$ is outside of the scope of this paper. Rather, we assume $\mathcal{B}$ to be given, reliable (i.e., it fits our purpose), and complete (i.e., all configurations, including features f and behaviors b, are known in advance and not discovered during the execution of the computation in $\mathcal{B}$). We also note that, as presented in the experiments in Section 6, $\mathcal{B}$ can be also generalized to include traditional data mining processes.

The feature collection model $\mathcal{F}$ then defines the feature collection process as a set of activities to be executed on a (class of) target system. It specifies how to collect evidence from smart devices and how to aggregate it in features, according to the CPS under evaluation and the features f specified in $\mathcal{B}$, as formally defined below.

Definition 3 ($\mathcal{F}$)**.** A feature collection model $\mathcal{F}$ is a pair $<\Phi,\mathcal{A}(\Phi)>$ where Φ describes the set of sequential flows of action ϕ_i undertaking the collection of relevant evidence e and $\mathcal{A}(\Phi)$ aggregates evidence e in features f. Each flow is composed of a set of activities $\phi_i=<t,a>$, where target t is either a function, procedure or data set, and a is the action to be executed on t to collect a specific e.

We note that features f can be retrieved as an aggregation of data and observable on smart devices (i.e., both application-specific and assurance-specific data flows in Figure 1). We also note that a feature collection model can specify complex flows

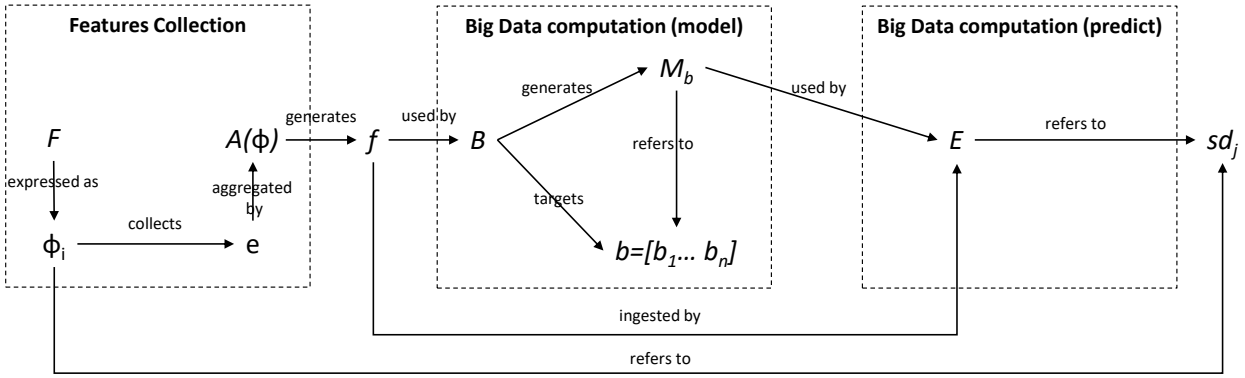

FIGURE 2 Policy template $\mathcal{P}^t$.

of action, for instance, requiring to access a system and inspect a configuration, to sniff traffic in specific locations, or to execute pre-processing at the edge of the network to produce the requested evidence.

Finally, $\mathcal{E}$ specifies how to annotate a smart device sd_j with a behavior b_i, by executing the behavioral model $\mathcal{M}_B$ according to features f as follows.

Definition 4 ($\mathcal{E}$)**.** Big Data computation $\mathcal{E}$ implements an evaluation process that takes as input the behavioral model $\mathcal{M}_B$ in $\mathcal{B}$ and the feature collection model $\mathcal{F}$, and returns as output a set of pairs (sd_j,b_i) where sd_j refers to a smart device and b_i to a behavior in $\mathcal{B}$. To this aim, the evaluation function first retrieves features f, then associates them to smart devices sd, and finally infers a behavior b according to the behavioral model $\mathcal{M}_B$ in $\mathcal{B}$.

We note that $\mathcal{E}$ annotates each smart device with a specific behavior, which will be used for evaluating its trustworthiness and the trustworthiness of its data, and can be also modeled as a traditional data mining process. Figure2 shows the template structure in a nutshell.

Example 3.1 (Assurance Policy Template $\mathcal{P}^t$)**.** Let us consider Example 2.1, with the scope of precisely positioning devices within the mall using RSSI. Let us consider an assurance policy template $\mathcal{P}^t$=<$\mathcal{B}$,$\mathcal{F}$,$\mathcal{E}$> that aims to verify the trustworthiness of the device for a given time window based on the human behavior. For simplicity, the assurance evaluation process uses the application-specific data flow only, meaning that there are no additional sources of information (assurance-specific data flow) that can be used to infer the behavior of interest. MAC addresses, RSSI, and AP SSID are collected using the specifications in $\mathcal{F}$ and filtered and aggregated as features f according to $\mathcal{B}$. Two human behaviors b_1 and b_2 are considered, namely, Line Of Sight (LOS) modeling correct behavior and No Line Of Sight (NLOS) modeling wrong behavior, respectively. The evaluation function $\mathcal{E}$ associates a behavior with each device for a given time window, according to a behavioral model $\mathcal{M}_B$ produced using the mean and variance of RSSI for each pair (MAC address,AP SSID).

3.2 | Assurance Policy Instance

An assurance policy template $\mathcal{P}^t$ is instantiated in an assurance policy instance $\mathcal{P}^i$ according to the specific scenario and system. Each policy instance is specified according to the technology available on the target system and contains *i)* the instances of the Big Data computation $\mathcal{B}$ and the collection model $\mathcal{F}$ in the template instrumented with real CPS endpoints, and *ii)* the instance of the Big Data computation $\mathcal{E}$ enriched with an evaluation function that is used to annotate each behavior b_i in $\mathcal{M}_B$ with a trust value. We introduce an instantiation function $\overset{I}{\rightarrow}$ that produces $\mathcal{P}^i$ as specialization of a given $\mathcal{P}^t$ as follows.

Definition 5 ($\overset{I}{\rightarrow}$)**.** Let $\mathcal{P}^t$=$\langle\mathcal{B},\mathcal{F},\mathcal{E}\rangle$ be an assurance policy template. An instantiation $\overset{I}{\rightarrow}$ is a transformation that takes $\mathcal{P}^t$ as input and produces $\mathcal{P}^i$=$\langle\mathcal{B}^+,\mathcal{F}^+,\mathcal{E}^+\rangle$ as output where:

- $\mathcal{B}\overset{I}{\rightarrow}\mathcal{B}^+$, where the technology-independent Big Data computation described in $\mathcal{B}$ is compiled in a technology-dependent version $\mathcal{B}^+$ ready to be executed on the target platform. $\mathcal{B}^+$ can be implemented using any language (e.g., Oozie) or

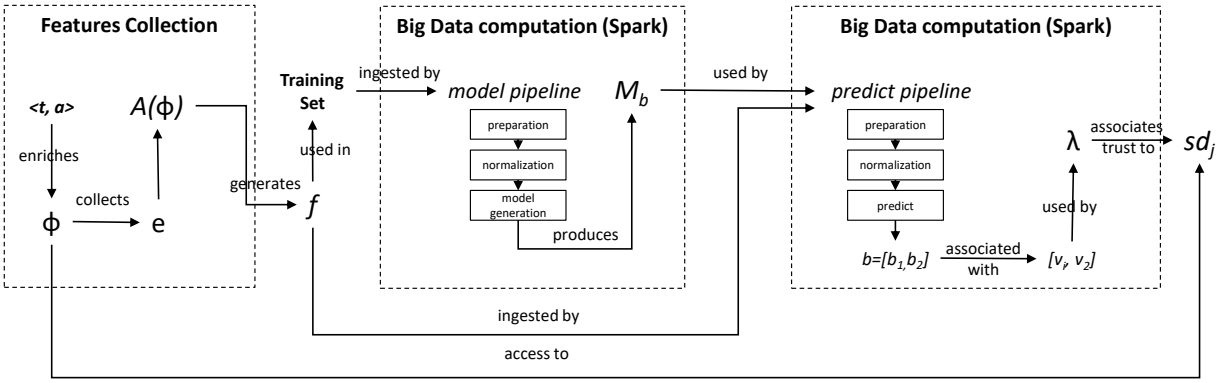

FIGURE 3 Policy instance $\mathcal{P}^i$.

platform (e.g., Apache spark) and specifies all configurations, end-points, features f, human behaviors b, training data set, analytics and processing modes (e.g., the adopted classification model).

- $\mathcal{F} \xrightarrow{I} \mathcal{F}^+$, where the aggregation function $\mathcal{A}(\Phi)$ is implemented and the feature collection model $\mathcal{F}$ is instrumented with real configurations, end-points, and activities. In particular, for each ϕ_i=<t,a>, target t is instantiated with the real function, procedure, or data set acting as a source of evidence, a is connected to the target to retrieve the evidence at the basis of feature generation.
- $\mathcal{E} \xrightarrow{I} \mathcal{E}^+$, where the Big Data computation $\mathcal{E}$ is instantiated in $\mathcal{E}^+$ similarly to $\mathcal{B} \xrightarrow{I} \mathcal{B}^+$. $\mathcal{E}^+$ is also enriched with a vector of trust values v_i, one for each behavior b_i, and an evaluation function λ that assigns a label $\lambda(b_{sd_j})$ to each smart device sd_j. $\lambda(b_{sd_j})$ corresponds to the trust value v modeling the trust of the specific smart device on the basis of the observed behavior b_{sd_j}.

We recall that the correctness of the instantiation of $\mathcal{B}$ ($\mathcal{E}$, resp.) in $\mathcal{B}^+$ ($\mathcal{E}^+$, resp.), as well as the process of configuring and training the behavioral policy, are out of the scope of this paper. Again, we use the approach in [13] to produce a technology-dependent Big Data computation $\mathcal{B}^+$ ($\mathcal{E}^+$, resp.) as a usable instance of the technology-independent Big Data computation $\mathcal{B}$ ($\mathcal{E}$, resp.) configured and trained to address the specific human behaviors (to annotate each behavior with a trust value, resp.). We note that the trust values in $\mathcal{E}^+$ cannot be associated with the behavioral model $\mathcal{M}_B$ before the instantiation process, since they could depend on the specific application scenario, for instance on the statistical reliability of the algorithm used to implement the behavioral model. Figure 3 shows the instance structure in a nutshell.

Example 3.2 (Assurance Policy Instance $\mathcal{P}^i$)**.** Following Example 3.1, the specifications in $\mathcal{F}$, $\mathcal{B}$, and $\mathcal{E}$ are instantiated with all configurations needed to interface with the CPS under evaluation in $\mathcal{F}^+$ (e.g., the local server collecting RSSI) and the Big Data platform in $\mathcal{B}^+$ and $\mathcal{E}^+$, respectively. $\mathcal{E}$ is also enriched with evaluation function λ and a vector of two trust values v_{LOS}=1 (trusted) and v_{NLOS}=0 (untrusted) associated with the two behaviors in $\mathcal{B}^+$ mapping LOS and NLOS scenarios, respectively.

3.3 | Assurance Policy Evaluation

The assurance policy evaluation is the final step of our methodology and consists of the execution of the assurance policy instance $\mathcal{P}^i$. It produces all artifacts for policy evaluation including the behavioral model $\mathcal{M}_B$, which is retrieved by executing $\mathcal{B}^+$ on the features f collected according to the feature collection model $\mathcal{F}^+$. The behavioral model and features are then used to evaluate the human behavior and, in turn, the trustworthiness of corresponding devices and collected data. In particular, the behavioral model $\mathcal{M}_B$ is first calculated by executing Big Data computation $\mathcal{B}^+$ and data collection model $\mathcal{F}^+$. Then, the Big Data computation $\mathcal{E}^+$ is executed as follows: *i)* each smart device sd_j is annotated with a behavior b_{sd_j} on the basis of behavioral model $\mathcal{M}_B$; *ii)* the evaluation function λ is executed according to Big Data computation $\mathcal{B}^+$, feature collection model $\mathcal{F}^+$, and trust values v in $\mathcal{P}^i$. The overall evaluation process returns as output a set of pairs (sd_j,$\lambda(b_{sd_j})$), where each single device is annotated with a trust value according to the selected behavior. This annotation is triggered any time data (e.g., batch for a given time window) are fed into the behavioral model.

Example 3.3 (Policy evaluation)**.** Following Example 3.2, evaluation function $\mathcal{E}^+$ is executed. Feature collection model $\mathcal{F}^+$ retrieves raw data on the devices, $\mathcal{B}^+$ calculates the behavior using RSSI variance according to data collected by $\mathcal{F}^+$, $\mathcal{E}^+$ returns a set of pairs (*MAC address, 1*) or (*MAC address, 0*) for the specific time window depending on the identified behavior. We note that discarding RSSI from untrusted devices in a given time window will improve the quality and reliability of the localization showing a positive impact on the computation.

4 | POLICY COMPOSITION

The policy template in Section 3.1 cannot model complex scenarios that are proper of CPSs. For instance, there are scenarios where alternative policies are composed depending on the type of data preparation, parallel policies are composed providing different types of analytics, or different policies need to be sequentially executed to evaluate trust at different layers. We then define an algebra of policy composition at the basis of a composite policy template $\mathcal{C}^t$. We consider three main operators, namely, sequence $\rightarrow$, alternative $\vee$, parallel $\wedge$. The syntax is defined according to the following BNF-like notation.

$$\mathcal{C}^t \ ::= \ \emptyset \mid \mathcal{P}^t \mid \mathcal{P}^t \rightarrow \mathcal{C}^t \mid \mathcal{P}^t \vee \mathcal{C}^t \mid \mathcal{P}^t \wedge \mathcal{C}^t$$

where $\mathcal{P}^t$ represents a policy template, while $\emptyset$ is an empty policy template. The above notation shows how a composite policy template $\mathcal{C}^t$ can be generated by iteratively applying the algebra operators. Our syntax is disambiguated by defining precedence and associativity rules for each operator. All operators are left associative, and $\vee$ and $\wedge$ have precedence over $\rightarrow$.

The composite policy template $\mathcal{C}^t$ is then instantiated in a composition policy instance $\mathcal{C}^i$ by iteratively applying the instantiation function in Definition 5 as follows. Let us consider $\mathcal{C}^t$ as a tree with maximum depth D, where each node is a policy template $\mathcal{P}^t$. For each depth d=0,…,D-1:

Step 1: all policy templates at depth d are instantiated in policy instances following Definition 5.

Step 2: a connector is added between sequential instances at depth d and instances at depth d+1. The role of the connector is to aggregate the results retrieved by the execution of the evaluation function of each policy instance at depth d and prepare the data to be analyzed by policy instances at depth d+1. Three scenarios are possible:

- $\mathcal{P}^i_j \rightarrow \mathcal{P}^i_{j+1}$, the connector receives as input the set of pairs $(sd_k, \lambda(b))$ returned by $\mathcal{P}^i_j$ as output and filters the data set according to the retrieved trust values $\lambda(b)$;
- $\mathcal{P}^i_1 \vee \ldots \vee \mathcal{P}^i_n$, the connector receives as input the set of pairs $(sd_k, \lambda_j(b))$ returned by the relevant policy $\mathcal{P}^i_j \in \mathcal{P}^i_1, \ldots, \mathcal{P}^i_n$ as output and filters the data set according to the retrieved trust values $\lambda_j(b)$;
- $\mathcal{P}^i_1 \wedge \ldots \wedge \mathcal{P}^i_n$, the connector receives as input n sets of pairs $(sd_k, \lambda_j(b))$ returned by $\mathcal{P}^i_j$, with j=1,…,n, as output, merge them according to a specific function (e.g., majority voting), and filters the data set according to the merged trust values $\lambda(b)$.

For each of the above scenarios, the filtered data set is finally returned by the connector as output and given as input to the subsequent node(s) in the composite policy instance $\mathcal{C}^i$.

Example 4.1 (Composite policy templates $\mathcal{C}^t$)**.** We present an example of composite policy for each of the three operators as follows.

1. $\mathcal{P}^t_1 \rightarrow \mathcal{P}^t_2$, where the output of $\mathcal{P}^t_1$ is given as input to another assurance process $\mathcal{P}^t_2$ that aims to distinguish between different behaviors. This is fundamental to select only those devices with behaviors that permits to calculate high-quality KPIs. To this aim, the data retrieved by executing $\mathcal{P}^t_1$ are filtered by removing untrusted MAC address and then given as input to $\mathcal{P}^t_2$ distinguishing between different behaviors.
2. $\mathcal{P}^t_1 \vee \mathcal{P}^t_2$, where the output of either $\mathcal{P}^t_1$ or $\mathcal{P}^t_2$ is returned as the output of $\mathcal{C}^t$. A composite policy composed of two alternative policies can be implemented to model scenarios where different data preparation must be implemented to accomplish privacy regulations. For instance, different anonymization techniques (e.g., General Data Privacy Regulation – GDPR in Europe) must be implemented to protect data collected from the smart devices of the users depending on the country in which the mall is built.

3. $\mathcal{P}_1^t \wedge \mathcal{P}_2^t$, where the output of $\mathcal{P}_1^t$ and $\mathcal{P}_2^t$ are merged in a single result. A composite policy composed of two parallel policies can be implemented to model scenarios where different data analytics are implemented to achieve better results. For instance, different types of analytics (e.g., SVM, k-means) are employed and their trust result averaged to obtain the final trust evaluation.

5 | DISCUSSION ON PRACTICAL APPLICABILITY

The practical applicability of our approach passes through the definition of a solution that can be seamlessly deployed in different environments with small changes to the trust assurance evaluation process. To this aim, our process supports high portability and reuse of the methodology, thanks to the generality of the policy template that can be instantiated in many different policy instances according to the considered CPS scenarios. The policy template is in fact technology- and domain-independent; it just models activities and procedures to be executed at an abstract level, while it does not specify how to implement them. The same template can therefore drive a range of different scenarios, whose peculiarities will be addressed during the instantiation process. The latter instantiates the template on the specific scenario and technology, producing a policy instance that is ready to be executed. Any changes of scenario/technology just impact the instantiation process but not the template, requiring to modify the way in which models are trained or data are collected. Policy evaluation step implements a continuous process that adapts to the evolution of the scenario and models captured by the policy instance.

A step further in this direction is represented by policy composition, where the portability of the policy template is extended to the composite policy template. The composite policy template, in fact, being technology- and domain-independent, can be ported to different domains and platforms, and later instantiated to accomplish the peculiarities of the specific scenario. As discussed in Section 4, the instantiation process independently instantiates each component template and then defines the connector(s) orchestrating their execution. As already discussed for single policy templates, any changes of scenario/technology just impact the instantiation process but not the policy template.

There is however a subtlety when policy template portability and policy composition are considered. Both approaches can be used to model scenarios where alternative deployments fit the requirements of a specific domain. However, alternative policy compositionmodels scenarios where different computations are carried out on (possibly) the same infrastructure. In this case, a different policy for each computation is needed specifying it in full details, rather than specifying a complex and unmanageable policy instantiation function (Definition 5), which differs case-by-case. Policy portability, instead, models scenarios where similar computations are carried out on different infrastructures.

Example 5.1. Let us consider our application scenario in Section 2.1 in two configurations as follows.

In the first configuration, we consider two malls (i.e., *mall1* and *mall2*) with two different settings. In *mall1*, *i)* APs can be locally accessed and provide information on MAC address, RSSI, and AP SSID, *ii)* AP position can be retrieved from the server. In *mall2*, APs can be remotely accessed and provide information on MAC address and AP SSID only. Given these two settings, there is the need of specifying a composite policy template $C^t = \mathcal{P}^t_{mall1} \vee \mathcal{P}^t_{mall2}$. Policy template $\mathcal{P}^t_{mall1}$ specifies a Big Data computation $\mathcal{B}_1$ using a supervised learning algorithm for classification (e.g., SVM), and a feature collection model $\mathcal{F}_1$ locally accessing MAC address, RSSI, AP SSID, and AP position. Policy template $\mathcal{P}^t_{mall2}$ provides a different collection model $\mathcal{F}_2$ that performs remote queries on the AP to collect MAC address and AP SSID, and a different computation $\mathcal{B}_2$ that implements a less sophisticated algorithm based on the frequency a MAC address appears in a given period of time. The behavior is based on the observed frequency of unique MAC addresses in a specific period of time; therefore, the instantiated computation $\mathcal{B}_2^+$ is simply based on finding the correct threshold that will be used in the classification process to identify the MAC address of customers and workers based on the frequency of occurrence, while the instantiated computation $\mathcal{B}_1^+$ does the same job using a SVM classifier. The instantiated collection model $\mathcal{F}_2^+$ has an empty aggregation due to the lack of precise AP location, and uses the set of detected MAC addresses as the only feature, while $\mathcal{F}_1^+$ has an aggregation of MAC, RSSI, AP SSID, and AP location providing a richer set of features to localize a devices within the mall.

In the second configuration, we consider the same policy template $\mathcal{P}^t$ instantiated according to two different infrastructures: the first based on a distributed Big Data platform, the second based on a monolithic, database management platform. We note that, in this case, a portable approach is used. The two instantiations just differ for the end point and technical configurations. The same algorithms are used (distributed as services in the first case, as jar files in the second case), and the same features are collected. This second example will be described in more details in the experimental evaluation in Section 6.

6 | EXPERIMENTAL EVALUATION

We experimentally evaluated our methodology in a real-world application scenario (Section 6.1) provided by Next Dev s.r.l. (Huko), a company offering Shopping Mall Analytics services and IoT devices for retails and malls. The goal of our experiments was to evaluate if and how much our assurance methodology improves on the quality and utility of the KPIs calculated in our application scenario, as well as the overall performance cost. This evaluation, being independent from the selected behavioral methodology, was run using a simple yet effective behavioral policy and analytics.

6.1 | Experimental Setup

Our experimental environment was a mall located in Chengdu, Cina, fully adopting the Huko's solution for Shopping Mall Analytics. The mall hosts 100 shops, each one equipped with a WiFi AP provided by Huko, able to capture mobile device probing on the WiFi network. APs communicates with the Huko server, continuously sending data (e.g., MAC address and RSSI) received from the mobile devices visiting the mall. The mall is visited by 130k people each month, 50k of them with Wifi activated and sensed by the Huko APs generating 25 GB/month of data on average. Huko's Shopping Mall Analytics engine computes KPIs on daily basis, meaning that the batch of data processed for KPI calculation is around 900 MB on average.

We considered a behavioral policy based on the idea that the ability of distinguishing different customers' behaviors makes the KPI computation more reliable. KPIs were computed on the basis of information of people tracked using AP signals (i.e., MAC addresses, RSSI and AP SSID). For simplicity, we considered two behaviors of interest *workers* and *customers*. Workers clearly taint the KPI computations, leading to inaccurate results when considered as customers visiting shops.

Huko's Shopping Mall Analytics engine was deployed on two different platforms: *1) Database Management Platform* using SQL-based technologies, *2) Big Data platform* based on Apache Spark. Since the same approach to behavioral modeling was used in both platforms, composing JAR files in the database management platform and REST services in Big Data platform, we designed a policy template $\mathcal{P}^t$, to be later compiled in two instances $\mathcal{P}^i_1$ and $\mathcal{P}^i_2$, respectively. In other words, the two instances $\mathcal{P}^i_1$ and $\mathcal{P}^i_2$ just differ on how the computation is implemented, while they share the same activities and collection model $\mathcal{F}$. The policy template $\mathcal{P}^t$ was defined as follows.

- The collection model $\mathcal{F}$ is composed of two sequential activities ϕ_1 and ϕ_2 capturing AP SSID, RSSI and MAC address of trusted devices and the AP location within the mall, respectively. It then aggregates them, using $\mathcal{A}(\phi_1, \phi_2)$, into a structured feature based on a join operation.
- The behavioral policy $\mathcal{B}$ defines a big data pipeline that produces the binary model that distinguishes between two behaviors, worker (b_1) and customer (b_2). It receives as input the aggregated data that refers to people movements retrieved by trusted devices only (see for more details $\mathcal{P}^i_1$ in Example 3.3) using the collection model $\mathcal{F}$.
- The evaluation function $\mathcal{E}$ evaluates each person's movements using a prediction pipeline based on the model produced by $\mathcal{B}$. It then associates the corresponding smart device sd_j (identified via the MAC address) with the proper behavior (either worker or customer).

We note that, except for some specific configurations such as evidence to be captured (i.e., MAC address, RSSI, AP, AP location), features to be used (i.e., aggregation of the evidence) and behaviors of interest (*workers* and *customers*), the policy template is generic and not strictly tailored to our experiments.

The template $\mathcal{P}^t$ is then instantiated in two policy instances $\mathcal{P}^i_1$ and $\mathcal{P}^i_2$. In the following, for conciseness, we only discuss $\mathcal{P}^i_2$.

- The collection model instance $\mathcal{F}^+$ specifies the two flows ϕ_1 and ϕ_2 with all configurations and endpoints, the first collecting MAC addresses, RSSI, and AP SSID from the local server storing data from all APs, and the second retrieving the location of a given AP SSID via direct connection to the server over SSH. The aggregation function $\mathcal{A}(\phi_1, \phi_2)$ implements a simple join operation on retrieved evidence to build the data set at the basis of the Big Data computation.
- Big Data computation $\mathcal{B}^+$ specifies a training pipeline as follows. Phase *data preparation* aggregates people movements into the mall, according to the MAC addresses and a specific training window, to be used as the classification features. Preparation may include algorithmic-specific data normalization or standard transformation for producing a data flow suitable for the binary model computation. Phase *model computation* generates the behavioral model. It ingests specific (labeled in our case) training data sources for the initial training. The binary classification is instantiated as a SVM classifier using Spark mlib. The generated model, retrieved by executing the training pipeline, is stored in HDFS.

```
INPUT
AppData: connectors to application-specific data flows
AssData: connectors to assurance-specific data flows
E_o:Set of objective evidence

OUTPUT
F: features data flow

MAIN F+
/*Log to AP location server*/
token = login(AssData.server, AssData.server.credentials);
/* retrieve AP locations */
loc = getLocation(token);
while (true) {
 /*Log to Edge server*/
 token = login(AppData.server, AppData.server.credentials)
 /* retrieve evidence */
 APEvid = SQLQuery(token, E_o)
 /* aggregate evidence */
 F = join(APEvid, loc)
 write(F)
}
```

```
INPUT
F_t: training set of features
W_d: dataset window dimension
W_g: grouping window dimension

OUTPUT
M: behavioral model
N_m: normalization model

MAIN B+
/* – Begin Big Data training pipeline – */
/* Data preparation*/
P = preparation(F_t, W_g, W_d);
/* Data normalization */
[P_n, N_m] = normalization(P, null);
/* Model computation */
M = classification(P_n);

PREDICTION(M, P)
/* SVM classification */
b=SVM.predict(M, P);
return b;
```

```
INPUT
M: behavioral model
N_m: normalization model
W_s: dataset window dimension
W_g: grouping window dimension
F: features data flow
V: trustworthy vector
sd: devices

OUTPUT
(sd_j,λ(b_sd_j)): labelled devices

MAIN E+
/* – Begin Big Data prediction pipeline – */
/* Data preparation*/
P = preparation(F, W_g, null);
/* Data normalization */
P_n = normalization(P, N_m);
/* Prediction */
b = prediction(M, P_n);
/* – End Big Data prediction pipeline – */
/* Device labelling */
(sd_j,λ(b_sd_j)) = labeling(sd_j,V,b)

PREPARATION(F, W_g, W_d)
/* Preparation query */
if (¬ isnull(W_d))
   F = horizontalCut(F, W_d);
P = dataCleaning(F);
P_g = dataAggregation(P, W_g);
P_g = SVMReshape(P_g);
return P_g

NORMALIZATION(P, N_m)
/* Normalize each feature in P*/
if isnull(N_m)
   N_m= normalizationModel(P);
P_n=SVMNormalization(P, N_m);
return [P_n, N_m];

CLASSIFICATION(P)
/* SVM classification */
M=SVM.fit(P);
return M;
```

FIGURE 4 An example of pseudocode for $\mathcal{P}_2^i$. Note that this pseudocode represents a Big Data pipeline in a generic format. It is not meant to be expressed with a specific Big Data language paradigm (e.g., Spark DAG).

- Big Data computation $\mathcal{E}^+$ defines a prediction pipeline for behavioral evaluation, where phases preparation and normalization are the same as in $\mathcal{B}^+$, but work on unlabeled data. The normalized data flow is finally used in phase *prediction* where a SVM predictor is used based on the model generated by $\mathcal{B}^+$ and stored in HDFS. $\mathcal{E}^+$ finally associates a trust value $\lambda \in \{0, 1\}$, where 0 corresponds to a worker and 1 to a customer, with device sd_j.

Figure 4 shows the pseudocode for the above policy instance $\mathcal{P}_2^i$.

The execution of the complete assurance evaluation process is finally orchestrated by the evaluation function $\mathcal{E}^+$ as follows. The continuous collection process specified in $\mathcal{F}^+$ is executed to collect and aggregate evidence into features. The collection process is then connected with the prediction pipelines. If behavioral model is empty, the training pipeline in $\mathcal{B}^+$ is executed on the training set to produce behavioral model $\mathcal{M}_B$. Once $\mathcal{M}_B$ is available, the prediction pipeline in $\mathcal{E}^+$ is executed. The training pipeline can continuously update the model, if needed, for instance when a sufficient amount of data (e.g., one month) is collected. The prediction pipeline exercises the available model, to predict the behavior of each person according to the selected

window of time (e.g., one day). The evaluation function finally associates a trust value with device sd_j according to the retrieved behavior.

6.2 | Performance, Accuracy, and Utility Evaluation

We experimentally evaluated the performance, accuracy, and utility of our approach executing the process instances $\mathcal{P}_1^i$ and $\mathcal{P}_2^i$ in the database management platform and Big Data platform, respectively. The database management platform is based on two Dell Poweredge R430 equipped with 32GB RAM and IntelXeon E5-2650 v4 2.2GHz as computing nodes and a DELL PowerVault MD3420 Full Flash as a storage node offering 1.9TB Solid State Drive. The entire infrastructure is virtualized on top of these servers using VMware ESXI tecnology, and installs a SQL Server-based platform executing our methodology. The Big Data platform is based on a Blade server PowerEdge M630 (VRTX) 2 x 687 Intel(R) Xeon(R) CPU E5-2620 v4 2.10GHz 192GB of RAM 120GB SSD. The entire infrastructure is virtualized on top of these servers using VMware ESXI tecnology, and installs an Apache-based Big Data platform based on Spark, executing our methodology.

We evaluated the *performance* of the Shopping Mall Analytics engine by measuring the computational time requested to execute the training pipeline based on SVM in $\mathcal{P}_2^i$ varying the data collection window (training set) in 2 days, 4 days, 1 week, 2 weeks, 3 weeks, 1 month. This choice was supported by two main reasons: *i)* the computational time requested for training pipeline dominates the whole execution time, *ii)* the training pipeline requires continuous updates to support effective processes (e.g., to prevent seasoning problems). We considered a strategy triggering a behavioral model update when half of the requested data are available, for instance, 2 weeks in case of models generated based on 1-month data. Figure 5 shows the performance of the training pipeline considering two distinct phases, preparation (including normalization) and behavioral model generation. We note that the behavioral model generation cost is much lower than the preparation cost. This is due to the fact that data must be condensed into daily journey before being used for the model generation; this requires to analyze the entire data set to find the occurrences of each MAC address and aggregate them on the basis of AP probing. Millions of records everyday must be aggregated according to hundreds of devices. We also note that while the computation time for phase preparation increases with the data set dimension (from a minimum of 15 minutes to a maximum of 140 minutes), the model generation is not increasing at the same rate (from a minimum of 1 minute to a maximum of 7 minutes). This effect is mainly due to the following reasons. On one side, the computational time of phase preparation is proportional to the number of records, while the computational time of phase behavioral model generation is proportional to the number of devices. On the other side, phase data preparation is a sequential activity based on query and aggregations requiring temporal dumps, while phase behavioral model generation can be easily parallelized in memory.

We then compared the performance of our approach in database management platform and Big Data platform. To this aim, we used the same behavioral approach based on SVM in both $\mathcal{P}_1^i$ and $\mathcal{P}_2^i$ and varied the data collection window in 1 week, 2 weeks, and 1 month to have a significant training set size for building the model. Considering $\mathcal{P}_1^t$ the overall computation performances for 1-month strategy is 92 hours compared to 2 hours and 27 minutes of $\mathcal{P}_2^t$. Similarly, $\mathcal{P}_1^t$ took 45 hours for 2-week strategy and almost 22 hours for the 1-week strategy, while $\mathcal{P}_2^t$ just 1 hour and 5 minutes and 35 minutes, respectively. As expected, the Big Data platform clearly outperforms the database management one, showing the negligible impact our methodology has on the overall performance. In both the cases the 1-week strategy showed the best performance.

We then evaluated the *accuracy* of our approach, which is almost the same with $\mathcal{P}_1^i$ and $\mathcal{P}_2^i$, since they use the same classification approach. We therefore evaluated the setting that best balances between performance and accuracy. To this aim, we considered the 1-month strategy as the ground truth data and computed the accuracy of the other strategies with reference to it. The 1-week strategy showed an accuracy of 40%, the 2-week strategy an accuracy of 65%. The 2-week strategy showed a substantial decrease in computational cost (more than 50%) still keeping a good level of accuracy (65% with reference to the 1-month strategy).

We then evaluated the *utility* of our approach using the 2-week strategy, with a model update every week. We first compared the quality of the KPIs using the approach without device trustworthiness evaluation and our approach based on assurance policy. The KPIs that Huko considered for the Chengdu mall at the time of this experimentation were *duration of visiting* and *visitor frequency* aggregated at mall level.[§] Even if the granularity of this KPI seems not suitable to obtain substantial advantages from the behavioral evaluation point of view, we obtained not negligible improvements with our methodology. For example, using the approach without the behavioral policy, the number of visits with duration greater than 180 minutes was computed as almost

[§]Other finer-grained KPIs currently available for other malls in EU will be available also for this one in the near future.

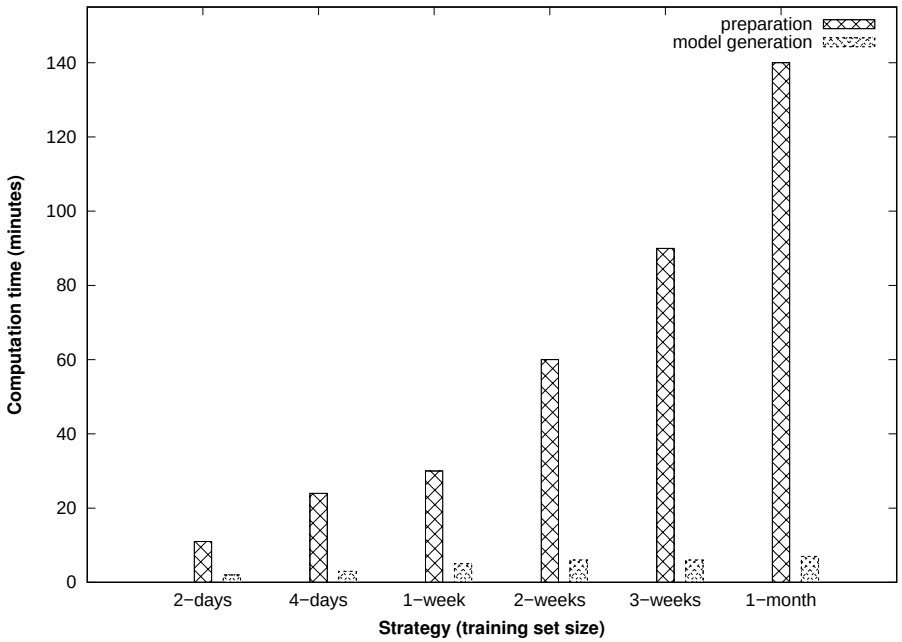

FIGURE 5 Computational cost of $\mathcal{P}_n^t$ varying the window from 2 days to one month.

13%, while with our methodology it decreased to 11.4%. As another example, using the approach without the behavioral policy, the 3.5% of visitor showed a frequency of visit greater than 4 times per month, which decreased to 1.5% with our approach.

7 | RELATED WORK

Behavioral analysis has been largely used in the area of machine learning for many purposes, such as Pacheco et al.[14] and Gupta et al.[15]. To the best of our knowledge, however, it has never been considered as a means to evaluate assurance of data in a human-centric CPS scenario. Only recently, the scientific community is starting drafting requirements on the importance of human behavior as an indicator of the trustworthiness of the data that human generates or manipulates[16], and in turn of the trustworthiness of decision processes based on them. This paper provides a first step in this direction by proposing a novel assurance methodology that selects trustworthy devices according to data on the behavior of device owners retrieved from the CPS environment.

The problem of evaluating the assurance of a given system, instead, is a time-honored research topic and has been under the attention of many researchers in recent years, mainly focusing on cybersecurity. Research on security assurance has followed the whole IT evolution initially providing approaches for software-based (e.g.,[17,18]) and service-based (e.g.,[19,20,21]) systems. More recently, the focus has moved to cloud-based systems, where assurance solutions employ testing and monitoring techniques to verify the behavior of a cloud-based system[5]. Di Giulio et al.[22] compare cloud assurance/security standards to evaluate how much they fulfill current threats to cloud assurance. Anisetti et al.[23] present a trustworthy test-based certification scheme for the cloud that supports continuous assessment, certification adaptation, and a chain of trust grounded on the certification authority requirements, and extend it in[11] to support continuous verification of model correctness against real and synthetic service execution traces to guarantee certification process soundness. Lins et al.[24] analyse whether certification can properly increase the user trustworthiness in using cloud services. Through signal theory, they identify which are the keys that most influence its effectiveness. Agarkhed et al.[25] and Hiremath et al.[26] describe a dedicated framework for data storage auditing in the cloud. Agarkhed et al.[25] provide auditing to check the correctness of outsourced data avoiding that information could be accessed by unauthorized users or hackers in the unsecured cloud network. Kirkman[27] proposes a solution to track and move data securely on the cloud by using smart contract and blockchain technology. Gonzales et al.[28] define cloud-trust, a security assessment model estimating the security level of cloud infrastructure through the analysis of high level security metrics quantifying the degree of confidentiality and integrity.

Our approach shares with the above assurance solutions the idea of capturing the evidence for assurance evaluation directly via testing (application-specific data flow in Figure 1) or indirectly via monitoring (assurance-specific data flow in Figure 1). It then extend these solutions towards CPS environments and IoT systems with the idea of targeting human behavior instead of machine behavior, and of providing a generic approach that goes beyond simple security evaluation. Assurance evaluation of IoT systems (and more in general of CPSs) is still in its infancy. The research community has first worked on providing security approaches for CPSs[29,30], and only in the last couple of years some preliminary assurance approaches emerged[31,32,33]. The difficulties in defining new approaches to IoT assurance are due to the fact that the peculiarities of IoT applications make existing assurance approaches for the cloud unusable. In fact, these approaches usually do not face the challenges in Section 2.1. More in details, they evaluate a system whose components are less numerous and heterogeneous, are not resource constrained, do not suffer from battery limitations, and are often under the control of known entities. Also, the system under evaluation shows a precise perimeter with known topology. Ardagna et al.[32] first discussed challenges in the design and development of assurance techniques for IoT, proposing a conceptual framework and architecture for IoT security assurance evaluation. Sato et al.[33] investigate the problem of trust establishment in IoT and propose an architecture for evaluating "*area-wise trust*", where the trust level considers device identification, monitoring of device behaviors, connection process to devices and connection protocols.

As stated above, the assurance evaluation of CPSs is complicated the active involvement of people in the execution of a specific business process. People can in fact use their devices to carry out some activities that are fundamental for the business process. Two users involvement are possible. On one side, we have Human-Provided Services[6,7], where people consciously distribute services as part of a wider business process. On the other side, as discussed in this paper, we have *Human-as-a-Sensor*, where people unconsciously distribute information through their device. The second scenario is standing in today systems, thanks to the proliferation of devices and the Internet of Things. Existing approaches consider behavioral analysis as a way to classify users and their activities, in order to improve the quality of service. Few approaches exist in literature and most of them insist in cybersecurity environment where user behavior is a well recognized impacting factor. Graitzer at al.[8] describe a predictive modeling framework based on a diverse set of data including inferred psychological/motivational factors. This framework is used to predict malicious insider exploits to trigger attention in a specific information analysis. Evans et al.[9] focus on the impact of human behavior in the area of cybersecurity assurance. The paper describes a list of behaviors of interest and some ideas on how to evaluate them based on standard qualitative and quantitative assessment methods (HEART). It also shows the advantage of having a behavior evaluation in cybersecurity. Lee at al.[10] consider the user behavior in the framework of an unusuality detection method based on geographical regularities obtained from a large number of geo-tagged tweets around Japan.

In this paper, we took a different approach and put forward the idea that behavioral analysis can be a means to evaluate the assurance of a specific device depending on the observed behavior of the user managing it. The idea is that the trustworthiness of a device, and in turn of the data generated through it, depend on how much the behavior of its user satisfies some requirements. Being able to annotate data with trust values permits to increase the quality of the processes and applications, which rely on such data, provided by CPSs. The proposed assurance approach has three main advantages: *i)* it does not impact on the device lifetime and resource consumption; *ii)* it manages heterogeneous devices in a single solution, providing a general-purpose approach based on human behavior; *iii)* it models the link between human behavior and device trustworthiness.

8 | CONCLUSIONS

The proliferation of IoT devices and the increasing availability of pervasive cyber-physical systems are radically changing the way in which people interact among them and with the surrounding environment. Today, people is just becoming another component of CPSs; they carry out tasks (either consciously or not), produce information, and stay connected virtually anywhere anytime. This scenario, which strongly binds people with technology, puts the safety of people at risk and then introduces an unprecedented need of guaranteeing a reliable environment. A reliable environment must be grounded on trustworthy data collection, cannot avoid people involvement, and need to accomplish those challenges introduced by IoT devices, which are heterogeneous, nomadic, resource constrained. In this paper, we presented a human-centric assurance solution, where the human behavior is used to identify trusted/untrusted devices and correctly filter out those data that might affect the quality of the overall CPS process. The behavior of each involved person is predicted according to a Big Data computation and monitored over time. The proposed approach has been tested in a real Chinese mall showing promising results.

References

1. Intel . *A Guide to the Internet of Things – Infographic*. Intel; : 2018. https://www.intel.com/content/www/us/en/internet-of-things/infographics/guide-to-iot.html, Accessed in date September 2018.
2. Statica . *Number of smartphone users worldwide from 2014 to 2020 (in billions)*. The Statistics Portal; : 2018. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/, Accessed in date September 2018.
3. Monsted B, Sapiezynski P, Ferrara E, Lehmann S. Evidence of complex contagion of information in social media: An experiment using Twitter bots. *PLoS ONE* 2017; 12(9).
4. Boshmaf Y, Muslukhov I, Beznosov K, Ripeanu M. The socialbot network: when bots socialize for fame and money. In: *Proc. of ACSAC*; 2011; Orlando, FL, USA.
5. Ardagna C, Asal R, Damiani E, Vu Q. From Security to Assurance in the Cloud: A Survey. *ACM CSUR* 2015; 48(1): 2:1-2:50.
6. Schall D, Truong HL, Dustdar S. Unifying Human and Software Services in Web-Scale Collaborations. In: Dustdar S, Schall D, Skopik F, Juszczyk L, Psaier H., eds. *Socially Enhanced Services Computing: Modern Models and Algorithms for Distributed Systems*Vienna: Springer Vienna. 2011 (pp. 17–27)
7. Moldovan D, Copil G, Dustdar S. Elastic systems: Towards cyber-physical ecosystems of people, processes, and things. *Computer Standards & Interfaces* 2018; 57: 76 - 82.
8. Greitzer FL, Hohimer RE. Modeling human behavior to anticipate insider attacks. *Journal of Strategic Security* 2011; 4(2): 3.
9. Evans M, Maglaras LA, He Y, Janicke H. Human behaviour as an aspect of cybersecurity assurance. *Security and Communication Networks* 2016; 9(17): 4667–4679.
10. Lee R, Sumiya K. Measuring geographical regularities of crowd behaviors for Twitter-based geo-social event detection. In: *Proc of SIGSPATIAL*ACM. ; 2010: 1–10.
11. Anisetti M, Ardagna C, Damiani E, Ioini NE, Gaudenzi F. Modeling time, probability, and configuration constraints for continuous cloud service certification. *Elsevier Computers & Security (COSE)* 2018; 72(Supplement C): 234 - 254.
12. Anisetti M, Ardagna C, Damiani E, Polegri G. Test-Based Security Certification of Composite Services. *ACM Transactions on the Web* 2018.
13. Ardagna C, Bellandi V, Bezzi M, Ceravolo P, Damiani E, Hebert C. Model-based Big Data Analytics-as-a-Service: Take Big Data to the Next Level. *IEEE Transactions on Services Computing (TSC)* 2018.
14. Pacheco J, Hariri S. Anomaly behavior analysis for iot sensors. *Transactions on Emerging Telecommunications Technologies* 2018; 29(4): e3188.
15. Gupta R, Salajegheh M, Christodorescu M, Sridhara V, Krishnamurthi G. Behavioral analysis to automate direct and indirect local monitoring of internet of things device health. 2018. US Patent 9,979,606.
16. ENISA . Cybersecurity Culture Guidelines: Behavioural Aspects of Cybersecurity. *Report of European Union Agency for Network and Information Security* 2018.
17. Damiani E, Ardagna C, Ioini NE. *Open Source Systems Security Certification*. Springer . 2009.
18. Herrmann D. *Using the Common Criteria for IT security evaluation*. Auerbach Publications . 2002.
19. Anisetti M, Ardagna C, Damiani E, Saonara F. A test-based security certification scheme for web services. *ACM Transactions on the Web (TWEB)* 2013; 7(2): 1–41.
20. Krotsiani M, Spanoudakis G, Mahbub K. Incremental Certification of Cloud Services. In: *Proc. of SECURWARE 2013*; 2013; Barcelona, Spain.

21. Zhang J. A Mobile Agent-Based Tool Supporting Web Services Testing. *Wireless Personal Communications* 2011; 56(1): 147–172.

22. Giulio CD, Sprabery R, Kamhoua C, Kwiat K, Campbell RH, Bashir MN. Cloud Standards in Comparison: Are New Security Frameworks Improving Cloud Security?. In: *Proc. of Cloud 2017*; 2017: 50-57

23. Anisetti M, Ardagna C, Damiani E, Gaudenzi F. A semi-automatic and trustworthy scheme for continuous cloud service certification. *IEEE Transaction on Services Computing (TSC)* 2017.

24. Lins S, Sunyaev A. Unblackboxing IT Certifications: A Theoretical Model Explaining IT Certification Effectiveness. In: *Proc. of ICIS 2017*; 2017; Seoul, South Korea: 1–13. keyword: Certification.

25. Agarkhed J, Ashalatha R. An efficient auditing scheme for data storage security in cloud. In: *Proc. of ICCPCT*; 2017: 1-5.

26. Hiremath S, Kunte S. A novel data auditing approach to achieve data privacy and data integrity in cloud computing. In: *Proc. of ICEECCOT 2017*; 2017: 306-310

27. Kirkman S. A Data Movement Policy Framework for Improving Trust in the Cloud Using Smart Contracts and Blockchains. In: *Proc. of IC2E 2018*; 2018: 270-273

28. Gonzales D, Kaplan JM, Saltzman E, Winkelman Z, Woods D. Cloud-Trust, a Security Assessment Model for Infrastructure as a Service (IaaS) Clouds. *IEEE Tansactions on Cloud Computing (TCC)* 2017; 5(3): 523-536.

29. Riahi A, Challal Y, Natalizio E, Chtourou Z, Bouabdallah A. A Systemic Approach for IoT Security. In: *2013 IEEE International Conference on Distributed Computing in Sensor Systems*; 2013: 351-355

30. Yang Y, Wu L, Yin G, Li L, Zhao H. A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal* 2017; 4(5): 1250-1258.

31. Zhang ZK, Cho MCY, Shieh S. Emerging Security Threats and Countermeasures in IoT. In: *Proc. of ACM ASIA CCS*ACM; 2015; New York, NY, USA: 1–6.

32. Ardagna CA, Damiani E, Schütte J, Stephanow P. A Case for IoT Security Assurance. In: Di Martino B, Li KC, Yang LT, Esposito A., eds. *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*Singapore: Springer Singapore. 2018 (pp. 175–192)

33. Sato H, Kanai A, Tanimoto S, Kobayashi T. Establishing Trust in the Emerging Era of IoT. In: *Proc. of IEEE SOSE*; 2016: 398-406.