An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments

Article

Accepted version

A. De Paola, P. Ferraro, S. Gaglio, G. Lo Re, S.K. Das

IEEE Transactions on Mobile Computing, 2016
DOI: 10.1109/TMC.2016.2599158

Publisher: IEEE

An Adaptive Bayesian System for Context-Aware Data Fusion in Smart Environments

Alessandra De Paola, Pierluca Ferraro, Salvatore Gaglio, Giuseppe Lo Re and Sajal K. Das

Abstract—The adoption of multi-sensor data fusion techniques is essential to effectively merge and analyze heterogeneous data collected by multiple sensors, pervasively deployed in a smart environment. Existing literature leverages contextual information in the fusion process, to increase the accuracy of inference and hence decision making in a dynamically changing environment. In this paper, we propose a context-aware, self-optimizing, adaptive system for sensor data fusion, based on a three-tier architecture. Heterogeneous data collected by sensors at the lowest tier are combined by a dynamic Bayesian network at the intermediate tier, which also integrates contextual information to refine the inference process. At the highest tier, a self-optimization process dynamically reconfigures the sensory infrastructure, by sampling a subset of sensors in order to minimize energy consumption and maximize inference accuracy. A Bayesian approach allows to deal with the imprecision of sensory measurements, due to environmental noise and possible hardware malfunctions. The effectiveness of our approach is demonstrated with the application scenario of the user activity recognition in an Ambient Intelligence system managing a smart home environment. Experimental results show that the proposed solution outperforms static approaches for context-aware multi-sensor fusion, achieving substantial energy savings whilst maintaining a high degree of inference accuracy.

Index Terms—I.2.6 Learning, J.9.f Wireless sensor networks

1 Introduction

Smart environments and intelligent systems designed for real-world applications are often based on the *sensing-reasoning-acting* paradigm [1]; generally, they exploit a sensory infrastructure to collect measurements, which is then used to obtain a high-level description of the state of the environment, and of the current context, reason about it, and select actions to be performed in order to achieve the desired system goals. In many cases the sensory infrastructure consists of a multitude of heterogeneous pervasive devices, which may produce a non-negligible uncertainty such as noisy data and measurements that impact the inference accuracy and energy consumption, and sensors on mobile devices, which are often energy hungry [2]. In such scenarios it is convenient to adopt a multi-sensor data fusion method, capable of dealing with such complex situations.

One of the most effective approaches for this purpose is the adoption of Bayesian belief networks [3], which exploit the statistical correlation between sensory measurements and the peculiarities of the surrounding world; they allow to deal with the heterogeneity of information sources and with the uncertainty in sensory data and developed models.

However, when performing data fusion, it might not always be efficient to sample all available sensors. On the contrary, if the sensory infrastructure is composed of devices with limited energy resources, it may be useful to activate only a subset of sensors, in order to increase the lifetime of the whole network. This is the case of smart environments whose pervasive sensory infrastructure includes wireless sensor networks (WSNs) [4]. WSNs are composed of devices, namely sensor nodes, characterized by programmability, wireless communications capability, and limited computational and energy resources, such that it is essential to extend the network lifetime by putting inactive nodes in stand-by state as long as possible. If the system exploits also sensors on mobile devices, their activation may cause a great energy consumption, thus it is convenient using them as data sources only when it is necessary. However, since different contexts may require different sensory capabilities, it is not desirable to determine a priori the subset of sensors to use. In a real-world scenario, the context conditions may change over time, implying the need for a system capable of dynamically selecting the subset of sensory devices.

1.1 Our Contributions

This work proposes an adaptive system performing multi-sensor data fusion based on a three-tier architecture, which allows the system to perform reasoning at different abstraction levels. Heterogeneous sensors at the lowest tier gather raw data about the environment. A dynamic Bayesian network at the intermediate tier performs the multi-sensor fusion and infers the context information necessary for the comprehension of the environment, such as the activities performed by the users in a smart environment. At the highest tier, a self-optimization subsystem reconfigures the sensory infrastructure, by selecting the subset of sensors to be used, in order to optimize its own performance. Such self-optimization process is performed through dynamic rules,

- *A. De Paola, P. Ferraro, S. Gaglio and G. Lo Re are with the DICGIM Department, University of Palermo, 90128 Palermo, Italy.*
  E-mail: {alessandra.depaola, pierluca.ferraro, salvatore.gaglio, giuseppe.lore}@unipa.it
  The work was done while Pierluca Ferraro was visiting Missouri University of Science and Technology.
- *Sajal K. Das is with the Department of Computer Science, Missouri University of Science and Technology, 65401 Rolla, MO, USA.*
  E-mail: sdas@mst.edu

so that the system is able to change its own goals in response to the context changes.

In order to validate the effectiveness of our approach, we considered the application scenario of an Ambient Intelligence (AmI) [1] system managing a smart home. AmI is an artificial intelligence application paradigm that focuses on the well-being of people and the satisfaction of their needs, through a pervasive infrastructure of sensors and actuators that surround users with a minimal degree of intrusiveness.

The above scenario is particularly appropriate for our evaluation, since frequent user interactions make the context highly dynamic. Furthermore, considering that sensory devices are usually only partially related to the observed phenomena, non-negligible noises are typically introduced. They may be caused by several different reasons [5], such as poor sensor quality, lack of sensor calibration, hardware failures, noise from external sources, and imprecision in computing derived values from measurements.

The capability of programming offered by the sensory devices also enables the management system to actually change the state of the sensory infrastructure. In such a scenario, our goal is to optimize the trade-off between the inference accuracy and energy consumption of sensory devices. The activities performed by users constitute the features to infer. Generally, activity recognition represents one of the key functionalities of an AmI system, since it enables an effective responsiveness toward the user needs.

Experimental study confirmed the capability of our approach to self-configure the sensory infrastructure by obtaining an optimal trade-off between the above two conflicting goals, thus achieving performances better than other possible static solutions. Our approach allows to go one step further toward real autonomous systems, capable of controlling the environment surrounding the users. The main novelty is its adaptiveness to dynamic environments and the capability of extending the lifetime of the sensory infrastructure as the context changes, thus reducing human interventions. Moreover, our solution has the capability of self-detecting those critical conditions in which a reconfiguration is needed in order to maintain a good quality inference.

1.2 Outline

The remainder of this paper is organized as follows. Section 2 briefly discusses the related work on multi-sensor data fusion and context-aware systems. Section 3 outlines the proposed architectural framework and provides a high level description of its modules. Section 4 describes the module that performs context-aware data fusion through a dynamic Bayesian network. Section 5 highlights the dynamic reconfiguration capabilities, explaining the behavior of the self-optimization modules. Section 6 presents the case study used to demonstrate the effectiveness of our approach with the help of activity recognition in an AmI scenario. Section 7 describes the experimental setting and the metrics used to evaluate the performance of the proposed approach, and analyzes our experimental results. Finally, Section 8 draws our conclusions with directions for future work.

2 Related Work

Multi-sensor data fusion techniques are widely used to meld data acquired by sensors deployed in the environment, in order to drive the process of knowledge abstraction from raw data and generating high-level concepts. Although plenty of research work has been done on data fusion, many challenging problems still remain unsolved, especially those inherent to the fusion of multi-source information, such as scalability, data inaccuracy and heterogeneity, outliers, or conflicting information. Further problems arise from the modeling of dynamic phenomena varying over time, or when addressing privacy and security concerns. For a comprehensive survey on the state of the art of multi-sensor data fusion solutions, please refer to [6].

Existing literature addresses the problem of data imperfection by proposing different approaches, based on various theoretical foundations. Probabilistic techniques [3], such as Naive Bayes classifiers, Hidden Markov Models, and Conditional Random Fields (CRFs), representing data uncertainty by using probability distributions, have been described in [7], which analyzes and compares the performance of these approaches in a smart home scenario. Alternative approaches deal with other aspects of data inaccuracy, such as vagueness, for example those based on the fuzzy set theory [8]. The authors in [9] proposed a general-purpose fuzzy logic architecture that combines multi-sensor data for automatic object recognition, control of system resources, and automated situation assessment. In [10], a multi-sensor image fusion for surveillance systems is proposed, which exploits fuzzy logic in order to enhance the fused image.

The problem of detecting sensory measurements that mismatch with the expected pattern of observed data has been thoroughly studied in the literature [11], with the aim of eliminating outliers from the fusion process. An adaptive, distributed Bayesian approach for detecting outliers in data collected by a wireless sensor network is proposed in [12] to guarantee reliability and fault tolerance, as well as to reduce energy consumption for unnecessary transmissions.

Heterogeneity of information sources is another significant challenge for data fusion systems. The raw data to analyze might be generated by a large number of heterogeneous sensors and extensive research effort has been devoted to coherently and accurately combine the resulting data [13], [14]. In recent years, a lot of attention has been paid to include context information in the data fusion process in order to reduce ambiguities and errors [15]. HiCon [16] is a hierarchical context aggregation framework that deals with a broad spectrum of contexts, from personal (e.g., the activities of individuals) to city-wide (e.g., locations of groups of people and vehicles) and world-wide (e.g., global weather and financial data). The authors in [17] defined a formal model capable of representing context and situations of interest, and developed a technique that exploits multi-attribute utility theory for fusing the modeled information and thereby attain situation awareness. An extensive framework to mediate ambiguous contexts in pervasive environments is presented in [18], [19].

When dealing with phenomena that evolve over time, *adaptiveness* is a fundamental feature. In such cases, where the external environment may constantly change, the system needs to dynamically adapt to the situation, and modify its behavior accordingly. Dynamic Bayesian networks (DBNs) [20], in addition to current sensory readings, consider the past belief of the system, thus capturing the dynamicity of the phenomena under observation. Several works adopt DBNs to perform adaptive data fusion for different applications, such as target tracking and identification [21], user presence detection [22], [23], user activity recognition [24], [25], and healthcare [26]. Other works integrate machine-learning algorithms in the data fusion process, to develop adaptive systems. Interesting examples of this trend are reported in [27], which exploits reinforcement learning, and in [28], which proposes kernel-based learning methods to improve the effectiveness of data fusion.

In the field of pervasive systems, and particularly in Ambient Intelligence, the adoption of an adaptive information fusion scheme, capable of exploiting context information, is fundamental to develop a truly smart environment [29] that is able to dynamically adapt to the external events. Context information, such as time, location, and user presence or activity, allows for refining the inference process, thus significantly improving the accuracy of reasoning [30]. The authors in [19] propose a resource-optimized framework for sensor networks based on DBNs and information-theoretic reasoning to minimize ambiguity in the context estimation process and context quality determination.

When the pervasive sensory infrastructure is enriched with mobile devices, as proposed in [31], the set of sensors capable of perceiving context information becomes larger. Nevertheless, those sensors are often energy hungry, thus many solutions in literature focus on the reduction of their consumption. In [32] a static scheme of exclusion of most costly sensors is proposed. SeeMon [33] is an energy-efficient context monitoring framework for mobile devices, which adopts event-based monitoring policies to save energy by reducing unnecessary wireless communications. ACE (Acquisitional Context Engine) [34] is a middleware for context-aware applications that dynamically learns associative rules among context attributes, so as to infer the value of expensive attributes by sensing cheaper ones. On a similar note, CARLOG [35] adopts a rule-based approach to minimize bandwidth usage, energy and latency, and supports multiple concurrent queries of context attributes, further minimizing bandwith usage.

The authors in [2] propose a solution that reduces energy consumption by selecting the set of sensors that achieves the minimum value of accuracy of estimation. The authors of [31] suggest instead to reduce the sensors sampling rate and propose a data fusion approach capable of dealing with such inhomogeneity of data sources.

Given the above considerations on the state of the art literature, this paper proposes a context-aware self-optimizing system for multi-sensor data fusion in pervasive computing scenarios, such as smart living environments. Differently from other previous works, our system focuses on dynamic management of sensors, and finding the best trade-off between inference accuracy and energy consumption of sensory devices. Furthermore, we exploit contextual information in a novel way, both to increase the accuracy of reasoning and to improve the adaptiveness of the system, thereby reducing the energy consumption.

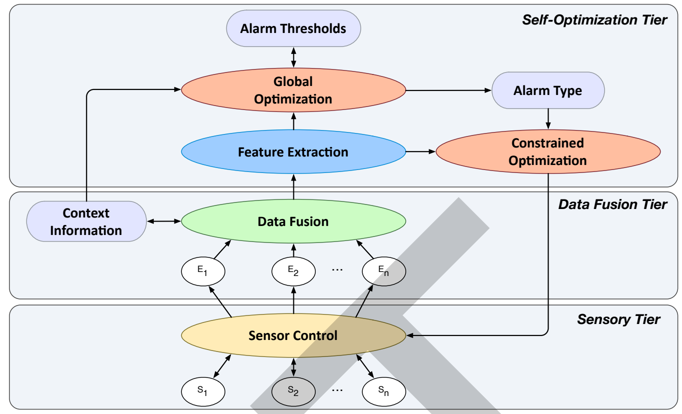


Fig. 1: Block Diagram of the three-tier architecture.

3 The Three-tier Architecture

We envision a three-tier architecture allowing the intelligent system to infer concepts related to the external world, from which events are observed by means of sensors. Furthermore, our architecture proposes a self-diagnostic component, in order to enable the system to reason about its own behavior and performance.

One of the main features of our approach is its capability of dealing with the inherent inaccuracy of sensors. Probabilistic techniques, such as dynamic Bayesian networks, enable the fusion of information coming from multiple sensors by explicitly modeling the noise and uncertainty of data so as to improve confidence and reliability of the reasoning process [6].

Another relevant feature of our system is the capability of adaptively selecting the best subset of sensors to be used in the data fusion process. Such selection is performed to activate only the sensors that are strictly necessary on the basis of the current needs of the system, rather than exploiting all the available data sources.

This approach meets several possible requirements of intelligent pervasive systems. It decreases the computational burden of the information fusion process, and consequently the system response time [21]. Moreover, it reduces the energy consumption of the sensory infrastructure; this aspect plays a relevant role in pervasive systems, such as Ambient Intelligence, considered here as a case study.

Although our approach minimizes the number of sensors used during the inference process, we require a high degree of accuracy. For this purpose, the system is designed to autonomously modify the state of the sensory infrastructure by switching sensors to a low-power mode which disables data gathering and transmission functionalities in order to minimize their energy consumption, whenever this does not compromise the inference accuracy.

The proposed architecture depicted in Fig. 1 consists of three different tiers characterized by increasing abstraction levels. The *lowest* tier (sensory) is in charge of the observation of external phenomena occurring in the environment (world) through the pervasive sensory infrastructure that manages only raw data. The *intermediate* tier (data fusion) copes with the uncertainty of gathered data and tries to infer the external world conditions, which contribute to define the current context, by performing a multi-sensor data fusion with the integration of further available context information. Finally, the *highest* tier (self-optimization) aims at finding an acceptable trade-off between costs (e.g., energy consumption) and system performance (e.g., accuracy of reasoning). Fig. 1 illustrates the main modules within each tier of the proposed architecture. The sensory and data fusion tiers present a simple structure exploiting respectively the following two functionalities:

- *Sensor Control*: collects raw data coming from the sensors, sends them to the *Data Fusion* module and modifies the state of sensory devices, by activating or deactivating them according to the policy indicated by the modules of the self-optimization tier;
- *Data Fusion*: exploits the available context information, performs probabilistic inference on the raw data coming from sensors, and combines them into describing the world through higher-level concepts.

The self-optimization tier exhibits the highest complexity, since it considers the state of the system and the current context conditions, and selects the actions to be performed in order to optimize the system behavior. The main functionalities of this tier are performed by the following three modules:

- *Feature Extraction*: selects the relevant context information about the external world and the internal state of the system in order to perform self-optimization. Such features are inferred both from the outcome of the *Data Fusion* module, and from the meta-analysis of the system internal behavior. The set of features is specific and depends on the particular application scenario. For this reason, in order to adapt our system to a specific scenario, it is necessary to identify the meaningful information needed for dynamically reconfiguring the sensory infrastructure.
- *Global Optimization*: exploits context information, in the form of features selected by the *Feature Extraction* modules, to dynamically modify the cost and uncertainty thresholds triggering the self-configuration performed by the *Constrained Optimization* module.
- *Constrained Optimization*: this module is activated by a set of alarms triggered by the *Global Optimization* module, when cost and uncertainty of the information fusion process exceed their respective thresholds. The cost and uncertainty quantities are two of the features selected by the *Feature Extraction* module for the specific scenario considered here.

A more accurate description of the role played by the most relevant modules is given in the following sections.

4 Multi-sensor Data Fusion

As the basis of the *Data Fusion* module we envisioned a dynamic Bayesian network (DBN), a special case of Bayesian networks, which is capable of capturing the dynamicity of the phenomena under observation, taking into account the past states in addition to the current observations. Differently from the most commonly used approaches for data fusion, which adopt Kalman filters [36] and hidden Markov models (HMMs) [37], we adopt the more general approach of the DBNs, in order to represent the influence of context variables on the state of the environment without any restrictions on the conditional probability distributions. DBNs allow us to model the time as slices representing the state of the world at a given instant, in addition to evidences representing the observable manifestation of the hidden world state. Since DBNs represent a good trade-off between expressiveness and tractability [20], they provide a useful tool for performing data fusion.

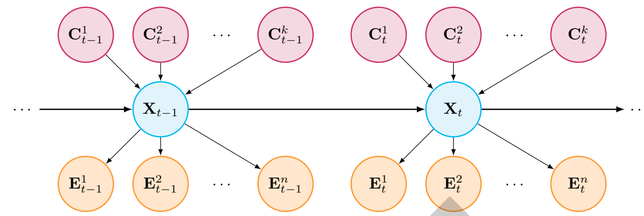


Fig. 2: The dynamic Bayesian network (DBN) used for performing the context-aware data fusion.

Fig. 2 sketches the DBN designed for our *Data Fusion* module. Its main goal is to infer the state of the world, in the form of a given feature of interest, which is represented by the hidden variable X_t. The belief update is performed on the basis of a set of sensory readings and a set of context information. Here $\boldsymbol{E}_t = (E_t^1, \ldots, E_t^n)$ is the set of sensory readings gathered by those sensors that are active in the time slice t, according to the policy of the *Constrained Optimization* module.

The set of context information, i.e., $\boldsymbol{C}_t = (C_t^1, \ldots, C_t^k)$, heavily depends on the application scenario; however, it is crucial to limit the number of context variables in order to control the size of the conditional probability tables (CPTs) and, consequently, the number of parameters to be learnt in the training phase.

The characterization of the DBN requires the definition of the *sensor model* and the *state transition model*. The probability distribution $P(\boldsymbol{E}_t|X_t)$ represents how sensory readings are affected by the current state of the system, and is named as the *sensor model*. The *state transition model*, i.e., $P(X_t|X_{t-1}, \boldsymbol{C}_t)$, expresses the probability that the state variable takes a certain value, given its previous value and the current context.

Since our DBN is a first-order Markov model, we can define the *belief* about a specific system state in the time slice t, i.e., x_t, as:

$$Bel(x_t) = P(x_t|\boldsymbol{E}_{1:t}, \boldsymbol{C}_{1:t}). \tag{1}$$

By following a procedure analogous to that adopted by authors of [38] for deriving the equation of Bayes filters, we obtain a practical formulation of our *belief*. In particular, by applying the Bayes rule, it is possible to express Eq. (1) as follows:

$$\begin{aligned} Bel(x_t) &= P(x_t|\boldsymbol{E}_{1:t}, \boldsymbol{C}_{1:t}) = P(x_t|\boldsymbol{E}_{1:t-1}, \boldsymbol{E}_t, \boldsymbol{C}_{1:t}) = \\ &= \eta \cdot P(\boldsymbol{E}_t|x_t, \boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) \cdot P(x_t|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}), \end{aligned} \tag{2}$$

where η is a normalizing constant. By using the Markov assumption, by considering that the sensor nodes in $\boldsymbol{E}_t$ do not depend on the context variables in $\boldsymbol{C}_t$, given the state variable X_t, and by assuming that sensor measurements are mutually conditionally independent, given the value of the parent node X_t, the following holds:

$$
\begin{aligned}
&P(\boldsymbol{E}_t|x_t, \boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) = P(\boldsymbol{E}_t|x_t, \boldsymbol{C}_{1:t}) = P(\boldsymbol{E}_t|x_t) = \\
&= \prod_{e_t^i} P(e_t^i|x_t),
\end{aligned}
\tag{3}
$$

where e_t^i is the specific value of the sensory reading gathered by the sensor i in the time slice t.

Moreover, the last term in Eq. (2) can be expressed as follows:

$$
\begin{aligned}
&P(x_t|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) = \sum_{x_{t-1}} P(x_t, x_{t-1}|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) = \\
&= \alpha \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}, \boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) \cdot P(x_{t-1}|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}),
\end{aligned}
\tag{4}
$$

where α is another normalizing constant. Now, $\boldsymbol{C}_t$ can be safely omitted from the last term, since X_{t-1} does not depend on the next context $\boldsymbol{C}_t$ if the next state X_t is not considered. Thus, using the Markov assumption, Eq. (4) can be expressed as:

$$
\begin{aligned}
&P(x_t|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t}) = \\
&\quad = \alpha \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}, \boldsymbol{C}_t) \cdot P(x_{t-1}|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t-1}) = \\
&\quad = \alpha \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}, \boldsymbol{C}_t) \cdot Bel(x_{t-1}).
\end{aligned}
\tag{5}
$$

Finally, by substituting Eq. (3) and Eq. (5) into Eq. (2), the belief can be defined with the following recursive formula:

$$
Bel(x_t) = \eta \cdot \prod_{e_t^i} P(e_t^i|x_t) \cdot \sum_{x_{t-1}} P(x_t|x_{t-1}, \boldsymbol{C}_t) \cdot Bel(x_{t-1}),
\tag{6}
$$

where α is integrated in the normalization constant η. Using Eq. (6), the inference can be performed by storing only two slices of the DBN, and thus the time and space required for updating the network belief are independent of the sequence length. The computational complexity of calculating Eq. (6) is $O(n+m)$, where n is the number of sensor nodes and m is the number of possible values of X_t. Thus, the overall complexity of computing $Bel(x_t)$ for all values of X_t is $O(m^2 + m \cdot n)$.

Since we already know the structure of the slices of our DBN, in the learning phase we have only to fill the CPTs. The method used to learn the CPTs may vary depending on the training set of historical data, Ξ, at our disposal. In a fully labeled dataset, we have only to compute sample statistics for each node. For example, let $\boldsymbol{P}_i$ denote the parents of a node V_i. The sample statistic $P(V_i = v_i|\boldsymbol{P}_i = \boldsymbol{p}_i)$ is given by the number of samples in Ξ having $V_i = v_i$ and $\boldsymbol{P}_i = \boldsymbol{p}_i$ divided by the number of samples having $\boldsymbol{P}_i = \boldsymbol{p}_i$. To learn the CPTs, we have to calculate these sample statistics for all the nodes in the network. If we don't have a fully labeled dataset, i.e., the values of one or more of the variables are missing for some of the training records, different techniques can be used, such as the Expectation Maximization (EM) algorithm or gradient ascent [3].

Example. We will now describe a running example illustrating how the DBN of the *Data Fusion* module operates.

TABLE 1: CPT for state transition: $P(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{C}_t^1)$.

		$\mathbf{X}_t$	
$\mathbf{C}_t^1$	$\mathbf{X}_{t-1}$	**0**	**1**
0	**0**	0.75	0.25
0	**1**	0.11	0.89
1	**0**	0.85	0.15
1	**1**	0.07	0.93
2	**0**	0.95	0.05
2	**1**	0.10	0.90

TABLE 2: CPTs for sensor models: $P(\mathbf{E}_t^1|\mathbf{X}_t)$, $P(\mathbf{E}_t^2|\mathbf{X}_t)$ and $P(\mathbf{E}_t^3|\mathbf{X}_t)$.

		$\mathbf{E}_t^1$			$\mathbf{E}_t^2$		$\mathbf{E}_t^3$	
		0	**1**	**2**	**0**	**1**	**0**	**1**
$\mathbf{X}_t$	**0**	0.50	0.20	0.30	0.87	0.13	0.68	0.32
	1	0.15	0.05	0.80	0.36	0.64	0.10	0.90

We consider a network with a single state variable $\mathbf{X}$, with two possible different values, three evidence nodes $\mathbf{E}^1$, $\mathbf{E}^2$, $\mathbf{E}^3$, taking 3, 2 and 2 values respectively, and a single context node $\mathbf{C}^1$, taking three possible values; the network is defined through the CPTs reported in Tables 1 and 2.

Let $P(\mathbf{X}_0) = \langle 0.85, 0.15 \rangle$ represent the probability distribution for the state variable at time $t = 0$, i.e., $P(\mathbf{X}_0 = 0) = 0.85$ and $P(\mathbf{X}_0 = 1) = 0.15$. If the sensory readings at time $t = 1$ are $[E_1^1, E_1^2, E_1^3] = [1, 0, 1]$, and the value of the context node C_1^1 is 0, the corresponding belief for the state variable can be computed according to Eq. (6):

$$
\begin{aligned}
&Bel(X_1 = 0) = \\
&\quad = \eta \cdot (0.2 \cdot 0.87 \cdot 0.32) \cdot (0.75 \cdot 0.85 + 0.11 \cdot 0.15) = \\
&\quad = \eta \cdot 0.037, \\
&Bel(X_1 = 1) = \\
&\quad = \eta \cdot (0.05 \cdot 0.36 \cdot 0.9) \cdot (0.25 \cdot 0.85 + 0.89 \cdot 0.15) = \\
&\quad = \eta \cdot 0.006,
\end{aligned}
$$

where the normalization constant can be computed as $\eta = \frac{1}{0.037+0.006} = 23.256$. Thus, the belief at time $t = 1$ is $Bel(X_1) = \langle 0.867, 0.133 \rangle$. We now suppose that at time $t = 2$ our sensors produce the readings: $[E_2^1, E_2^2, E_2^3] = [2, 1, 0]$, and that the value of the context node C_2^1 is 1. Applying again Eq. (1), we obtain $Bel(X_2) = \langle 0.606, 0.394 \rangle$. As mentioned, the context information greatly influence inference results. Indeed, if the value of the context node C_2^1 had been 0 at time $t = 2$, the corresponding belief would have been $Bel(X_2) = \langle 0.507, 0.493 \rangle$.

5 Context-aware Self-Optimization

The self-optimization modules aim to dynamically optimize the state of the sensory infrastructure, in order to find the best trade-off between performance and costs. To this end they need to exploit the maximum of available information on the external world and on the internal system conditions. Useful information can be obtained by the analysis of the behavior of the *Data Fusion* module, in addition to the other information acquired from the external world.

The *Feature Extraction* module produces a feature vector containing meta-information about the world state and the system state, which, as a whole, defines the current context.

Such information are then exploited by the *Global Optimization* and *Constrained Optimization* modules to update the alarm thresholds and reconfigure the sensory infrastructure respectively, as will be described in the following.

5.1 Feature Extraction

The *Feature Extraction* module produces for each time slice t a feature vector composed of the following elements:

$$FV_t = [\boldsymbol{conf}_t, \boldsymbol{WSB}, T_E, RUE, SC, \boldsymbol{SIE}], \tag{7}$$

where $\boldsymbol{conf}_t$ is the configuration of the sensory infrastructure, $\boldsymbol{WSB}$ is the belief distribution about the world state as inferred by the *Data Fusion* module, T_E is the elapsed time since the last estimated transition of the world state, RUE is an estimation of the reasoning uncertainty, SC is the cost associated to the active sensors, and $\boldsymbol{SIE}$ is an estimation of the importance of each sensor for the inference.

The sensory configuration, which fully describes the state of each sensor, is defined as a binary vector, $\boldsymbol{conf}_t = [s_t^1, s_t^2, \ldots, s_t^n]$, where $s_t^i \in \{0,1\}$. Each element s_t^i specifies whether the corresponding sensor E^i is *off* or *on*, respectively, in the time slice t.

The *belief* about the state of the world, $\boldsymbol{WSB}$, is obtained directly from the *Data Fusion* module at each time slice, as calculated by Eq. (6). The *Global Optimization* module exploits such information along with T_E, the elapsed time since the last transition of the world state as perceived by the system. If we define the most probable world state as:

$$x_t^* = \arg\max_{x_t}\big(Bel(x_t)\big), \tag{8}$$

the *Feature Extraction* module evaluates the number of time slices elapsed since the last transition of the world state with the following equation:

$$T_E(t) = \begin{cases} T_E(t-1)+1 & \text{if } x_{t-1}^* = x_t^*, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

The uncertainty and cost indices associated with a specific sensory configuration are used by the *Global Optimization* module to decide whether to trigger an alarm, thus requiring a reconfiguration of the sensory infrastructure. To measure the uncertainty of the probabilistic reasoning performed by the *Data Fusion* module, we propose to adopt an index based on the definition of Shannon entropy [39]:

$$RUE \stackrel{def}{=} -\sum_{x_t} Bel(x_t)\log_2(Bel(x_t)). \tag{10}$$

Assuming that the cost of acquiring information from node E^i in the time slice t is expressed by the function $f_{cost}(E_t^i)$, the total cost associated to a specific sensory configuration can be evaluated as:

$$SC \stackrel{def}{=} \sum_{E_t^i \in \boldsymbol{E}_t} f_{cost}(E_t^i)\cdot s_t^i. \tag{11}$$

The specific cost function closely depends on the application scenarios; we defer the description of the one adopted for our case study to Section 6.

Finally, in order to support the system to select the most useful sensors to activate at any time slice, we adopted a heuristic that estimates the importance of each sensor in the current situation, based on the Kullback-Leibler (KL) divergence [40], known as relative entropy or *information gain*. The KL divergence of an approximate distribution G, with regard to a *true* distribution F, measures the information that is lost when G is used instead of F. It is defined as:

$$D_{KL}(F\|G) = \sum_i F(x_t)\log\bigg(\frac{F(x_t)}{G(x_t)}\bigg). \tag{12}$$

The information gain of an active sensory configuration can be computed as the KL divergence of the prediction $P(X_t|\boldsymbol{E}_{1:t-1}, \boldsymbol{C}_{1:t})$ with regard to the belief $Bel(X_t)$.

However, if we consider a sensory infrastructure with n sensors, in order to exactly determine the optimal sensory configuration, the optimization modules should evaluate the information gain for all the possible 2^n sensor states, with an effort that, as n grows, becomes quickly intractable. For this reason, we define a heuristic which roughly approximates the information gain of a given sensory configuration with the simple sum of the information gains of each sensor. The current information gain of a single active sensor can be simply evaluated by comparing the belief obtained by using its last reading with the belief obtained by ignoring such evidence. We define the belief that would be obtained by ignoring the evidence e_t^i, for a specific state of the world, as a function of the belief $Bel(x_t)$:

$$\overline{Bel}_{e_t^i}(x_t) = \beta\cdot\frac{Bel(x_t)}{P(e_t^i|x_t)}, \tag{13}$$

where β is a normalizing factor. We can then compute the information gain of a single active node E^i, in the time slice t, as the KL divergence of $\overline{Bel}_{e_t^i}(X_t)$ with regard to $Bel(X_t)$, as follows:

$$\begin{aligned} inf\_gain(E_t^i) &= inf\_gain(E_t^i = e_t^i) = \\ &= D_{KL}(Bel(X_t)\|\overline{Bel}_{e_t^i}(X_t)). \end{aligned} \tag{14}$$

The evaluation of the information gain of an inactive sensor shows a higher computational cost, since the absence of a reading involves the computation of the belief change for all the possible sensory readings. Namely, the information gain of an inactive sensor should be calculated as follows:

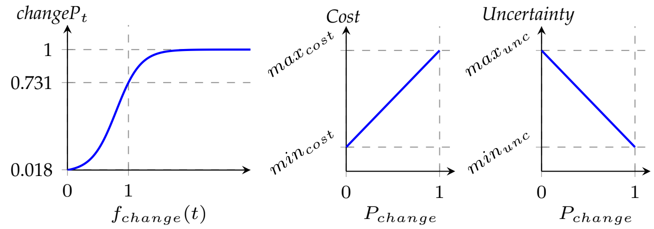
$$inf\_gain(E_t^i) = \sum_{e_t^i} P(E_t^i = e_t^i)\cdot inf\_gain(E_t^i = e_t^i). \tag{15}$$

However, in order to reduce the computational cost, in our heuristic we adopted a further simplification that exhibits the advantage of considering not only the instantaneous information gain, but also the past history. Accordingly, we do not exploit Eq. (15) to compute the information gain of an inactive sensor; rather, at each time slice, we evaluate the current information gain only of the active sensors and update their sample mean and variance. In this way, we can consider an estimate of the value of information provided by the whole sensory configuration $\boldsymbol{conf}_t$ as,

$$\boldsymbol{SIE} \stackrel{def}{=} [inf\_gain_{avg}(\boldsymbol{conf}_t), inf\_gain_{var}(\boldsymbol{conf}_t)], \tag{16}$$

where

$$\begin{aligned} inf\_gain_{avg}(\boldsymbol{conf}_t) &\stackrel{def}{=} \sum_i inf\_gain_{avg}(E_t^i)\cdot s_t^i, \\ inf\_gain_{var}(\boldsymbol{conf}_t) &\stackrel{def}{=} \sum_i inf\_gain_{var}(E_t^i)\cdot s_t^i. \end{aligned} \tag{17}$$



(a) $changeP_t(f_{change})$ function, with $\lambda = 5$ and $\nu = 4$.

(b) θ_{cost_t} and θ_{unc_t} functions.

Fig. 3: The functions used in the *Global Optimization* module.

Since the KL divergence does not satisfy the triangle inequality, our approach constitutes only a heuristic, although it performs well in practice, as will be shown in the experimental section. Moreover, storing the sampling mean and variance of the information gain for n sensors is much more efficient than exploring all the 2^n sensory configurations and all possible readings of inactive sensors.

The computational complexity of calculating the feature vector FV_t is dominated by the evaluation of Eq. (17). The information gain of a single sensor node can be derived in $O(m)$, where m is the number of possible values of X_t. Their mean and variance can be updated online in $O(1)$, thus, the overall complexity of the operations performed by the *Feature Extraction* module with n sensors is $O(m \cdot n)$.

5.2 Global Optimization

The goal of the *Global Optimization* is to dynamically drive the behavior of the self-configuring component, according to the context changes. Static criteria regarding reconfiguration frequency or global objectives of the system are inappropriate for dynamic scenarios.

The *Global Optimization* module dynamically modifies the alarm thresholds, concerning inference uncertainty and cost, used for triggering and constraining the reconfiguration performed by the *Constrained Optimization* module, as depicted in Fig. 1. The basic idea is that, in order to obtain a significant energy saving, it is appropriate to reduce the set of active sensors when it is expected a reduced variability in context conditions, i.e., no change in the state of the world is expected, even if such an action may involve a reduction in the inference accuracy. On the other hand, whenever significant context variability is expected, a better policy could be to increase the sensory activity of the system. To this end, the *Global Optimization* module uses the estimated temporal lengths of world states, as learned during the training phase by the *Data Fusion* module, the inferred belief about the current state of the world, and the time elapsed since the last state transition observed by the system.

For instance, if the system identifies a state condition with a high probability to persist for a long time, it might switch *off* most of the sensors without a relevant accuracy reduction. Conversely, if the system estimates a sudden transition for current state, it should activate all the sensors necessary to achieve the desired accuracy, even at the cost of higher energy consumption.

To this end, the *Global Optimization* module tries to predict if the state of the world will change in the next time slice, by estimating the probability of a state condition transition with the following soft-threshold function:

$$changeP_t(f_{change}) = 1 - \frac{1}{1 + e^{\lambda f_{change}(t) - \nu}}, \tag{18}$$

where

$$f_{change}(t) = \frac{elapsedTime_t}{expectedDuration(X_t)}, \tag{19}$$

and λ and ν are parameters of the soft-threshold function. This implies the probability of transition of a state condition increases as $f_{change}(t)$ value grows, as shown in Fig. 3a. The function $expectedDuration(X_t)$ in Eq. (19) is computed as the expected value of the current state duration, as follows:

$$\begin{aligned} expectedDuration(X_t) &= \sum_i w_t^i \cdot avgDuration(x^i), \\ w_t^i &= Bel(x_t^i), \end{aligned} \tag{20}$$

where $avgDuration(x^i)$ represents the estimation of the temporal length of a specific state of the world x^i, as learned during the training phase by the *Data Fusion* module, and each weight w_t^i is the belief of a specific state in the time slice t, such that $\sum_i w_t^i = 1$.

The transition probability of the state condition computed by the *Global Optimization* module is used to dynamically modify the cost and uncertainty thresholds, which can vary between minimum and maximum values set by the system administrator. Whenever the uncertainty and cost of the inference performed by the *Data Fusion* module exceed these thresholds, an alarm is triggered and consequently the *Constrained Optimization* module performs the reconfiguration of the sensory infrastructure. The thresholds for cost and uncertainty, i.e., θ_{cost_t} and θ_{unc_t}, are modified according to the following equations:

$$\begin{aligned} \theta_{cost_t} &= min_{cost} + \\ &\quad + (max_{cost} - min_{cost}) \cdot changeP_t(f_{change}), \\ \theta_{unc_t} &= min_{unc} + \\ &\quad + (max_{unc} - min_{unc})(1 - changeP_t(f_{change})). \end{aligned} \tag{21}$$

As shown in Fig. 3b, when $changeP_t(f_{change})$ is close to 0, the system enters a power-saving mode, thereby reducing the cost threshold and raising the uncertainty threshold. Conversely, when $changeP_t(f_{change})$ is close to 1, the system gives priority to the accuracy of the inference, raising the cost threshold and reducing the uncertainty threshold.

At each time slice, the *Global Optimization* module updates the thresholds and compares them with the cost and uncertainty computed by the *Feature Extraction* module, checking whether any alarm is triggered. If both the thresholds are exceeded, priority is given to the reduction of uncertainty, since the main goal of the system is to infer the state of the world in a precise and reliable way.

The computational complexity of the *Global Optimization* module is dominated by Eq. (20), which can be computed in $O(m)$, where m is the number of possible values of X_t.

The *Global Optimization* module is implemented as an Influence Diagram [3], a generalization of a Bayesian network, capable of supporting probabilistic decision-making. Such Influence Diagram is shown in Fig. 4. It merges two types of information: (i) information describing the state

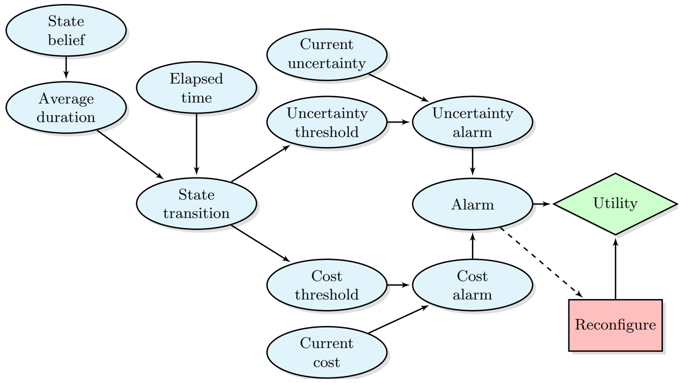


Fig. 4: The *Global Optimization* module Influence Diagram.

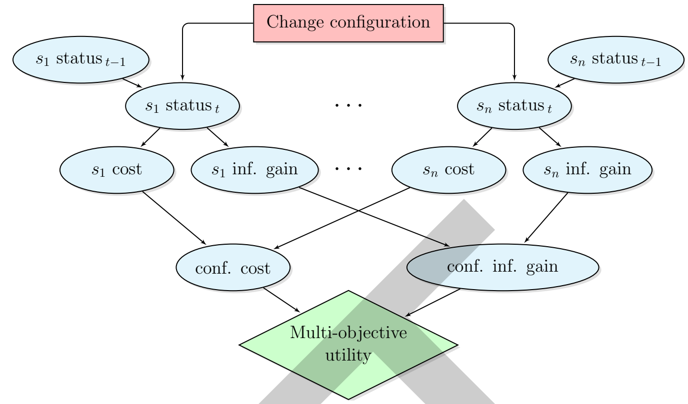


Fig. 5: The *Constrained Optimization* module multi-objective influence diagram.

of the world, captured by the *Data Fusion* module, and (ii) meta-information about the behavior of the *Data Fusion* module, extracted by the *Feature Extraction* module. All such information represents the context from the point of view of the *Global Optimization* module.

The *Alarm* node and the *Reconfigure* decision node represent binary variables that indicate, respectively, whether an alarm is triggered and whether the system should dynamically modify the state of its sensors.

The conditional probability distributions of the *State transition* node and of the thresholds nodes are based on Eq. (18) and Eq. (21), respectively. Finally, the utility function of the Influence Diagram is computed as the *XNOR* (exclusive NOR) of the *Alarm* variable and the *Reconfigure* decision variable, in order to trigger a reconfiguration when an alarm condition holds.

5.3 Constrained Optimization

In order to achieve the best possible trade-off between cost and reasoning accuracy, the *Constrained Optimization* module leverages the dynamic thresholds set by the *Global Optimization* module to dynamically modify the state of the sensory infrastructure. For instance, it can reduce the energy consumption of the sensory infrastructure at the cost of a sacrifice in accuracy, by switching redundant devices to the low-power mode, thus suspending their data gathering and transmission functionalities, as explained in Section 3.

To manage the above conflicting goals, we adopt a multi-objective trade-off analysis based on a Pareto-dominance criterion [23], [41], since the traditional approach of using multi-attribute utility theory [17], capable of maximizing a single expected utility that summarizes all the considered attributes, has several drawbacks. Namely, it is often difficult to formulate an utility function which correctly models the optimization problem. Moreover, such an approach would require a precise assessment of the relative relevance of the goals to achieve.

In order to guarantee a steady adaptation of the sensory infrastructure, and to limit the computational complexity of the optimization, we allow only *atomic* reconfigurations, i.e., actions that modify the state of single sensors. Thus, in a system with n sensors, the *Constrained Optimization* module chooses among n possible sensory configurations when an alarm is raised. If neither of the thresholds is exceeded by the system, no reconfiguration will be performed. The cost and information gain of each achievable configuration can be derived in constant time. Thus, the overall complexity of computing these quantities for all n achievable configurations is $O(n)$.

The *Constrained Optimization* module categorizes the sensory configurations achievable in the time slice t as two disjoint classes: *dominated* configurations and *non-dominated* configurations. A sensory configuration $\boldsymbol{conf}_t^*$ is *non-dominated*, or Pareto-optimal, if no other solution has better values for all the objectives considered, so that the following holds $\forall i \in \{1, \ldots, n\}$:

$$\begin{cases} cost(\boldsymbol{conf}_t^*) \leq cost(\boldsymbol{conf}_t^i) \\ \quad\quad\text{or} \\ inf\_gain_{avg}(\boldsymbol{conf}_t^*) \geq inf\_gain_{avg}(\boldsymbol{conf}_t^i). \end{cases} \tag{22}$$

In order to determine the action that achieves the best trade-off between costs and performance, the *Constrained Optimization* module considers only the set of *non-dominated* configurations, also known as Pareto front. The selected sensory configuration depends on the alarm raised by the *Global Optimization* module, as follows:

- if only the cost threshold was exceeded, the sensory configuration minimizing the expected $cost(\boldsymbol{conf}_{t+1})$ is selected among the *non-dominated* solutions;
- otherwise, if only the uncertainty threshold or both thresholds were reached, the sensory configuration maximizing the expected $inf\_gain(\boldsymbol{conf}_{t+1})$ is selected among the *non-dominated* solutions.

The Pareto front can be derived in $O(n^2)$ time, and the final selection of the configuration to choose requires $O(n)$ time, which yields a total complexity of $O(n^2)$. The operations performed by the *Constrained Optimization* module are implemented as a multi-objective influence diagram [42]. This approach extends the traditional influence diagrams, which only allow a single utility node or a single combined utility node that is sum or product of other utility functions [43]. To overcome this limitation, a new kind of utility node was defined in [42] as a vector of objectives, together with an algorithm for the solution of such multi-objective influence diagrams.

We propose the influence diagram shown in Fig. 5, where the *Change configuration* decision node indicates the single sensor whose state must be changed, depending on its previous state. Finally, the best sensory configuration is chosen among the optimal front, as explained above, and a control message is sent to the appropriate sensor, thus triggering the dynamic reconfiguration of the sensory infrastructure.

6 Case Study: Activity Recognition in AmI Scenario

In order to prove the effectiveness of our approach, we chose an Ambient Intelligence (AmI) environment as an application scenario. In particular, we simulated the implementation of our solution in a smart home system enriched with a pervasive sensory infrastructure based on WSNs and sensors on mobile devices. Such scenario is suitable for the experimental evaluation for several reasons: first of all, AmI contexts are characterized by a high dynamism, due to their continuous and unpredictable interaction with users. Second, the use of WSNs and sensors on mobile devices as sensory infrastructure poses severe constraints on device energy consumption, justifying the necessity of finding the best trade-off between inference accuracy and energy consumption of sensory devices. Finally, since observed data are not collected by specialized sensors, but rather by sensors whose readings are only partially correlated with the environmental features of interest [23], it is necessary to perform a multi-sensor data fusion.

Our system is designed for inferring any hidden characteristic of the observed world. In the case study proposed here, we chose to consider the activities performed by the user as context features to be inferred, since such information allows AmI systems to be fully responsive to user needs. Moreover, recognizing user activities such as eating, cooking, sleeping, or working [7], [44], is one of the major challenges of AmI systems.

The approaches to address this challenge vary greatly depending on the kind of activities to classify, the data fusion method adopted, and the sensors used. For example, inertial sensors such as accelerometers and gyroscopes, also installed on mobile devices, are commonly used to recognize activities that involve physical movements, e.g., walking, running, sitting down and standing up [45], [46]. The authors of [45] compared the advantages and disadvantages of single-sensor and multi-sensor wearable systems, and proposed an approach based on a decision tree classifier to find a trade-off between recognition accuracy and computational and communication complexity. In recent years, several works, such as [46], have considered the possibility of exploiting the increasing pervasiveness of smartphones to recognize user activities, by merging the raw data coming from the sensors embedded in these devices. The data collected by wireless sensors which are pervasively deployed in the environment are often used to recognize a wider range of Activities of Daily Living (ADLs), such as eating, cooking, sleeping, or working [7], [44].

The implementation of our system in a specific application scenario implies the characterization of (i) the components of the feature vector which are related to the state of the world, (ii) the context information acquired directly from the external world, and (iii) the cost function adopted by the self-configuration modules.

In the scenario considered here, i.e., user activity recognition in a smart home, the feature vector contains the belief about the activity currently performed by the user and the time elapsed since the last activity transition, in addition to features which do not depend on the application scenario.

The context factors we chose to refine the inference performed by the *Data Fusion* module are the period of day, i.e., morning, afternoon, evening or night, the month, and the day of the week. These proved to be the best available context information in our experimental setting for improving the accuracy of the system.

We defined the cost function with the goal of enforcing an energy saving policy that aims to extend the devices' lifetime; thus, we propose a cost function whose value increases when the residual battery charge of the corresponding sensor decreases, according to the following equation:

$$f_{cost}(E_t^i) = baseCost(E_t^i) \cdot \left(1 + \gamma \, \left(1 - charge(E_t^i)\right)\right) , \tag{23}$$

where $0 \leq charge(E_t^i) \leq 1$ is the remaining charge of sensor E^i in the time slice t, and γ is a parameter that controls the relative importance of the charge level. When the value of γ increases, the system is more inclined to use all the sensors in a uniform manner, instead of preferring only the best ones, and this greatly extends the sensory infrastructure lifetime with little impact on accuracy, as will be shown in the next section.

7 Experimental Evaluation

7.1 Simulation Setting

In order to evaluate the performance of our system, we simulated its behavior in a smart home where several programmable wireless sensor nodes were deployed and where the interaction of the user's mobile device with the home Wi-Fi network reveals the presence of the user at home. The traces collected by a set of real sensors have been exploited in order to simulate the interaction of sensory devices with the environment, according to principles defined in [47]. For the sake of simplicity, we simulated the energy consumption of sensory devices by assuming a constant energy cost for each data transmission. The ground truth about user activities and the corresponding sensory traces were obtained from the Aruba dataset of the CASAS Smart Home Project [7] developed at Washington State University. The Aruba dataset contains annotated data collected in a smart apartment with a single resident, over a period of seven months. In such dataset, events are generated by motion sensors, door sensors, and temperature sensors.

A preprocessing of the original data was required to test our system. In particular, we partitioned the sequence of sensor events into regular time slices, counting how many times each sensor was activated during each slice. It is therefore important to carefully select an appropriate length for time slices; otherwise, many slices will not contain any sensor event; we selected a time slice length of 30 seconds.

Moreover, a heuristic was developed to label each interval with the predominant activity performed by the user during that time slice. The Aruba dataset annotated eleven activities of daily living (ADLs), namely *Bed to Toilet*, *Eating*, *Enter Home*, *Housekeeping*, *Leave Home*, *Meal Preparation*, *Relax*, *Resperate*, *Sleeping*, *Wash Dishes*, and *Work*. We added a new activity, named *Outside*, that takes into consideration the periods of time when the user is not inside the smart home, i.e., the intervals between the *Leave Home* and the *Enter Home* activities. This information can be used by the system to further optimize the energy consumption of the sensory infrastructure, since all the sensors installed in the smart home can be deactivated when no one is present, with the exception of the door sensors, thus minimizing energy consumption without sacrificing inference accuracy. In a real scenario, it is very difficult, if not impossible, to predict all the activities that will be performed by users and, furthermore, a fixed list of activity classes cannot take into consideration the unavoidable transitions between activities. To address both of these problems, we added a further activity class to the ones annotated within the original dataset, named *Other*, as suggested in [44]. This special class groups all the sensor events that do not match any of the known activity classes. Nearly 20% of the sensor events in our dataset belong to the *Other* class, and therefore we believe it is essential to detect it correctly in a real world system. However, considering the heterogeneity of the activities grouped by this class, it is very challenging to recognize it with good accuracy, and many approaches in the literature simply ignore it, relying solely on a list of predetermined activities, as noted in [44]. In the following, the experimental results obtained by our system are presented by considering the *Other* class as well as ignoring it.

7.2 Performance Metrics

The experimental results compare the performance of three different systems. The first one, called *All-On*, does not exploit the optimization modules to adaptively self-configure its behavior at runtime, and uses all the available sensors in each time slice, thus minimizing the uncertainty of reasoning regardless of energy consumption. The second one, called *Subset-On*, gives priority to energy savings, leveraging only a small subset of sensors, i.e., 10 of the 39 available sensors, and does not exploit the optimization modules to improve its performance. The last system, finally, is the context-aware self-optimizing system proposed in this work, and it fully exploits the *Global Optimization* and *Constrained Optimization* modules described in the previous sections. The first two systems are considered as baseline for comparison with the proposed one.

The most common metric to evaluate activity recognition systems is the average accuracy, defined as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \tag{24}$$

where *TP*, *TN*, *FP*, and *FN* are, respectively, the true positives, true negatives, false positives and false negatives. However, accuracy alone is not sufficient to evaluate different approaches, since data are skewed towards the most probable activities. For this reason, we adopted additional metrics to provide more detailed analyses of the performance of the systems. In particular, we determined the *precision* (positive predictive value), as fidelity measure, and the *recall* (sensitivity), for measuring completeness, which are defined as follows:

$$\begin{aligned} precision &= \frac{TP}{TP + FP}, \\ recall &= \frac{TP}{TP + FN}. \end{aligned} \tag{25}$$

Precision and recall, in turn, are used to calculate the *F-score*, which is a very important metric to evaluate activity recognition systems, as stated in [44]. *F-score* is defined as the harmonic mean of precision and recall, as follows:

$$F\text{-}score = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{26}$$

In order to evaluate the different approaches, we used the cross validation method, dividing the dataset into ten parts. For each test, nine parts are used for learning system parameters and the tenth is used for the test. This process is repeated changing the test set. We compared the uncertainty, cost, and accuracy of each system, as well as precision, recall, and F-score of each activity, as detailed in the following.

7.3 Experimental Results

The first experiment presented in this section is the comparison of uncertainty, cost, and accuracy trends of the three systems during a given week, when also the *Other* activity class is considered. In this first test, we discarded the temperature readings, since we noticed a low correlation between such data and the activities performed by the user. Fig. 6 shows the simple moving averages (SMAs), i.e., the unweighted mean of the values collected in the previous hour. As expected, the *All-On* system exhibits the lowest uncertainty and the highest energy cost. Conversely, the *Subset-On* system presents the highest uncertainty and a lower power consumption than the *All-On* system. The *Adaptive* system shows an uncertainty close to that of the *All-On* system, with a lower power consumption. It is worth noticing that the average cost of the *Adaptive* system increases very slowly over time, since the optimization modules are smart enough to exploit sensors in a uniform manner, so as to maximize the life span of the WSN and minimize power consumption. With regards to the accuracy, the *All-On* and *Adaptive* systems perform similarly, and occasionally the *Adaptive* system overcomes the *All-On*. The *Subset-On* system, on the other hand, behaves poorly, and its accuracy is behind those of the other two systems.

Table 3 reports the average accuracy of the two baseline systems compared with the proposed *Adaptive* system, in each test of the cross-validation experiment. Table 4 summarizes the results of the cross-validation tests, reporting the average accuracy (both considering and excluding the *Other* activity class), uncertainty, power consumption, number of active sensors, and F-score of the three systems in all tests. It can be observed that the proposed *Adaptive* system achieves an accuracy slightly higher than that of the *All-On* system. This may seem a bit unexpected, but it can be explained by considering the particularity of the dataset used, which mainly contains motion sensors. In such a scenario, using an optimal subset of sensors can lead to higher accuracy

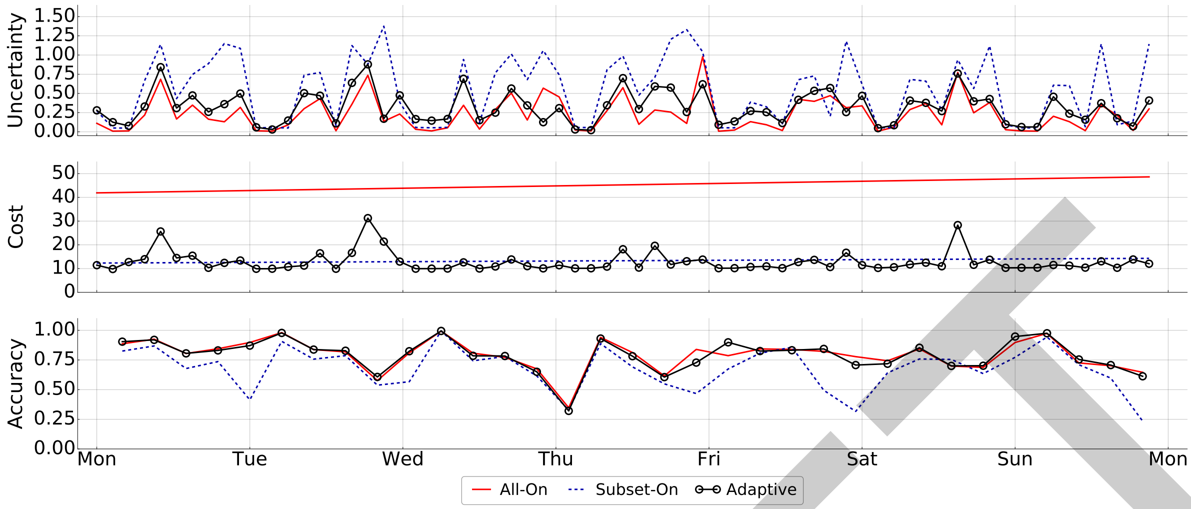


Fig. 6: Uncertainty, cost, and accuracy trends of the three systems considered during a given week.

TABLE 3: Average accuracy of the two baseline systems compared with the proposed *Adaptive* system, in each test of the cross-validation experiment based on the CASAS dataset [7].

All-On	Subset-On	Adaptive
0.760	0.659	0.755
0.802	0.704	0.818
0.788	0.670	0.796
0.792	0.677	0.803
0.810	0.672	0.815
0.803	0.657	0.811
0.790	0.539	0.777
0.782	0.652	0.788
0.811	0.687	0.817
0.769	0.644	0.773

TABLE 4: Cross-validation results reporting the average accuracy, uncertainty, power consumption, number of active sensors, and F-score of the two baseline systems compared to the proposed *Adaptive* system.

	All-On	Subset-On	Adaptive
Accuracy	0.791	0.656	0.795
Accuracy without *Other*	0.882	0.705	0.878
Accuracy with temperature	0.749	0.534	0.786
Uncertainty	0.251	0.573	0.349
Power Consumption	44.771	13.168	13.494
Active Sensors	34.000	10.000	12.124
F-score	0.421	0.271	0.408
Execution Time	1.00x	0.76x	0.81x

than keeping all sensors on, since it allows the system to exclude some false positive readings. For instance, when the user is sleeping, the *Adaptive* system exploits only sensors near the bedroom, while the *All-On* system will leave all the sensors on, possibly incurring in a greater number of false positive readings coming from sensors in other rooms, which may have a non-negligible noise level. The *Subset-On* system shows a lower accuracy, i.e., almost 14% less than the *Adaptive* system. The average costs of the two approaches, on the other hand, are very similar, with only a 1% difference. The average cost of the *All-On* system is more than three times that of the proposed *Adaptive* system.

Table 4 confirms that the uncertainty of the *Adaptive* system is comparable to that of the *All-On* system, whilst the *Subset-On* system performs worse than the other two, with an average uncertainty that is 64% higher compared to that of the proposed system. Similar considerations apply to the F-score of the three systems, which is similar for the *All-On* and *Adaptive* system, and more than 30% lower on the *Subset-On* system. To evaluate the optimization overhead, Table 4 also reports the average execution time, normalized by the same factor, so as to have 1 computation unit in the case of the *All-On* system. We notice that the *Adaptive* system gets a 19% speed-up with regard to the *All-On* system. Thus, the speed-up due to the fact of using fewer sensors in the data fusion process overweights the optimization overhead. Considering that the *Adaptive* and *Subset-On* systems use a similar number of sensors, on the average (12 and 10, respectively), the small difference in execution time of the two systems (0.81x and 0.76x, respectively) gives an idea of the optimization overhead.

These results indicate the advantage of using the *Adaptive* approach, since it performs better than the baseline systems, finding an optimal trade-off between performance and consumption. As expected, the accuracy of all systems improves significantly if the *Other* activity class is ignored, increasing by almost 10% in the *All-On* and *Adaptive* systems, and by about 5% in the *Subset-On* system. Table 4 also shows the accuracy of the systems when using temperature data. In the case of the *Subset-On* system, the five temperature sensors were added on top of the fixed 10 normally used. We can notice that the accuracy of all systems decreases when using temperature data, since there is a low correlation between this information and the activities performed by the user. However, the accuracy of the *Adaptive* system decreases by less than 1%, whilst the other two systems show a more noticeable reduction, since the *Adaptive* system detects the low importance of the temperature sensors, and often turns them off.

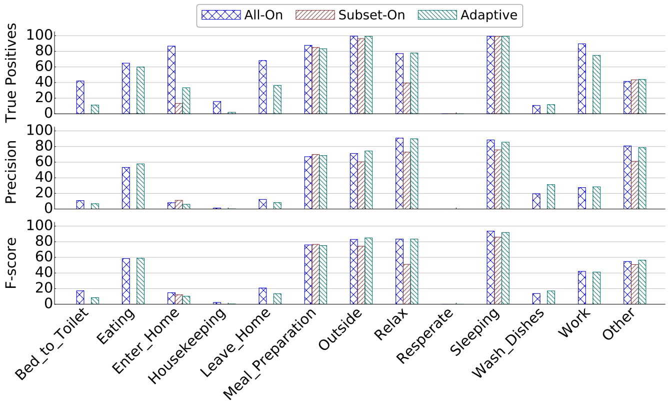


Fig. 7: True positives, precision and F-score of each activity.

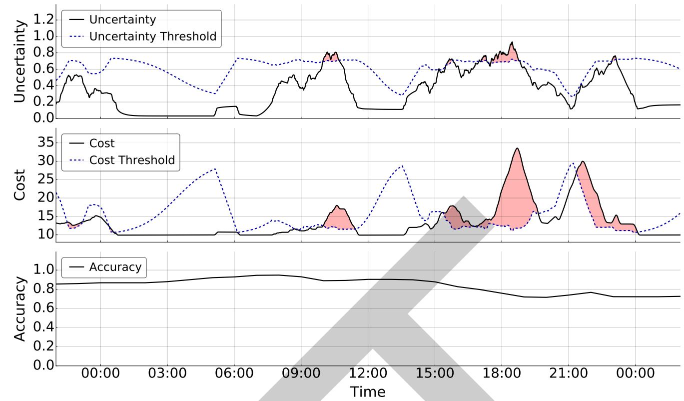


Fig. 8: Uncertainty, cost, and accuracy trends of the *Adaptive* system in a single day, showing the adaptive thresholds.

We also compared the true positives rate, precision, and F-score of each activity in the three considered systems, without considering temperature sensors, as shown in Fig. 7. The performance of the *All-On* and *Adaptive* systems regarding many activities is largely comparable, and the *Adaptive* approach actually obtains better results than the *All-One* system in some of them. However, there are a few activities that are better recognized by the *All-On* system, due to a very low average duration that does not allow the *Global Optimization* module to correctly estimate the probability of an activity transition. As expected, the performance of the *Subset-On* system is visibly worse than the other two approaches. Nevertheless, it can be observed that some activities are easily recognized by all the systems, i.e., *Sleeping*, *Outside* and *Meal Preparation*, while other activities are difficult to handle, regardless of the approach considered, i.e., *Housekeeping* and *Wash Dishes*. This can be explained by considering that some activities are performed in well-defined locations and times during the day, and therefore are better recognized using only motion sensors, while other activities are less structured, and some heterogeneous sensors should be installed in order to recognize them in a satisfying manner.

The experimental evaluation includes also a deep analysis of the effect of the *Global Optimization* module on the behavior of the *Adaptive* system. Fig. 8 shows the uncertainty and the cost trends of the *Adaptive* system, with the related alarm thresholds, during a given day. The uncertainty and cost alarms triggered by the system in the same day are also highlighted. As explained in Section 5.2, when the *Global Optimization* module expects a reduced variability in context conditions, it changes the alarm thresholds in order to switch off most of the sensors without a relevant reduction of accuracy, switching them on when it believes that the current activity will change in a short time, so as to ensure adequate accuracy, even at the expense of higher energy consumption. *Sleeping* is the most regular activity in our dataset, and it is also the one with the longest average duration. This regularity makes the *Sleeping* activity ideal to analyze the way in which the system dynamically changes the alarm thresholds. The cost threshold is low at the beginning of the night, when the user goes to sleep, whilst the uncertainty threshold is high. This means that the system believes as unlikely a context transition in a short time, and thus enters a power-saving mode, reducing the cost threshold and raising the uncertainty threshold. During the night, the probability of an activity transition, as defined in Eq. (18), increases, and therefore the system gradually reduces the uncertainty threshold and increases the cost threshold. In the morning, when the probability of an activity transition is close to 1, the system gives priority to the accuracy of the inference, thereby increasing the cost threshold to its maximum value, and reducing the uncertainty threshold to its minimum.

Finally, we compare the effectiveness of using the variable cost function, defined in Eq. (23), against an approach that leverages a fixed cost function, that does not increase when the residual battery of the corresponding sensor decreases. The goal of choosing a variable cost function is to enforce an energy saving policy that extends the sensory infrastructure lifetime by using sensors in a uniform manner. Fig. 9 compares the two approaches, showing the number of time slices in which each sensor has been powered on during the simulations, and the remaining energy charge at the end of the simulations. In this experiment, we also included the temperature sensors, so as to show that the system correctly avoid sensors with a very low information gain, which are not deemed useful. It is evident that with a variable cost function the system uses sensors more uniformly than the fixed cost approach, greatly extending the sensory infrastructure lifetime. However, it is worth noticing that, despite the energy policy, sensors which show a very low information gain, such as the temperature sensors (i.e., the last five sensors in Fig. 9), are seldom used by the system. Finally, comparing the accuracy of the variable cost approach against that of the fixed cost approach, we observe a decrease of less than 1%.

8 Conclusions

In this paper, we proposed and evaluated a context-aware self-optimizing adaptive system for sensory data fusion. The system is based on a three-tier architecture. At the lowest tier, the inference subsystem leverages dynamic Bayesian networks for inferring the state of the world, exploiting contextual information to increase reasoning accuracy. At the same time, two self-optimization modules are responsible for determining the subset of sensors to use at each

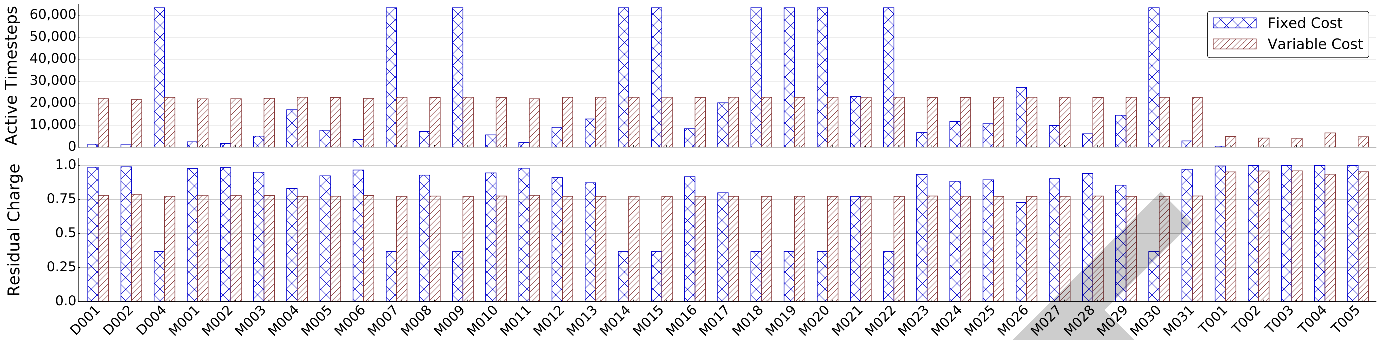


Fig. 9: Sensor statistics usage with a fixed and variable cost function. The x axis enumerates all the available sensors: door sensors (Dxxx), motion sensors (Mxxx) and temperature sensors (Txxx).

time slice, finding an optimal trade-off to minimize energy consumption and maximize sensing accuracy.

As case study for evaluating the proposed system we chose the activity recognition problem in an Ambient Intelligence scenario. The experimental evaluation confirmed that using all the available sensors is not a good strategy, both in terms of energy efficiency and computational burden. In a real world scenario, context evolves constantly, and the system has to dynamically adapt to the current situation, by reconfiguring its own sensory infrastructure. The results of our experiments support these statements, showing that the proposed adaptive system performs better than static systems, achieving substantial energy savings compared to a system that statically uses all the available sensors, with only a small increase in inference uncertainty.

In the current study, both training and test data are collected from the same environment. As part of future development, we are interested in evaluating the generalization potential of the proposed approach by considering training and test data coming from different smart homes or offices. Furthermore, we plan to test the system in a real scenario with heterogeneous sensors, including data coming from wearable devices such as smart watches.

9 Acknowledgments

The authors are grateful to the anonymous reviewers for insightful comments and constructive suggestions that helped us improve the quality of the manuscript significantly.
The work of S.K. Das is partially supported by the following NSF grants: IIS-1404673, CNS-1404677, CNS-1545037, and CNS-1545050.

References

[1] D. Cook, J. Augusto, and V. Jakkula, "Ambient Intelligence: technologies, applications, and opportunities," *Pervasive and Mobile Computing*, vol. 5, no. 4, pp. 277–298, 2009.

[2] A. Rahmati, C. Shepard, C. Tossell, L. Zhong, and P. Kortum, "Practical context awareness: Measuring and utilizing the context dependency of mobile usage," *IEEE Trans. Mobile Computing*, vol. 14, no. 9, pp. 1932–1946, 2015.

[3] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[4] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless Sensor Network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[5] D. Wang, L. Kaplan, and T. F. Abdelzaher, "Maximum likelihood analysis of conflicting observations in social sensing," *ACM Trans. Sensor Networks (ToSN)*, vol. 10, no. 2, p. 30, 2014.

[6] B. Khaleghi, A. Khamis, F. O. Karray, and S. N. Razavi, "Multisensor data fusion: a review of the state-of-the-art," *Information Fusion*, vol. 14, no. 1, pp. 28–44, 2013.

[7] D. J. Cook, "Learning setting-generalized activity models for smart spaces," *IEEE Intelligent Systems*, vol. 2010, no. 99, p. 1, 2010.

[8] L. A. Zadeh, "Fuzzy sets," *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[9] J. Stover, D. Hall, and R. Gibson, "A fuzzy-logic architecture for autonomous multisensor data fusion," *IEEE Trans. Industrial Electronics*, vol. 43, no. 3, pp. 403–410, Jun 1996.

[10] Y. Zheng and P. Zheng, "Multisensor image fusion using fuzzy logic for surveillance systems," in *Proc. 7th Int'l Conf. on Fuzzy Systems and Knowledge Discovery (FSKD)*, vol. 2, 2010, pp. 588–592.

[11] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for Wireless Sensor Networks: a survey," *Communications Surveys & Tutorials, IEEE*, vol. 12, no. 2, pp. 159–170, 2010.

[12] A. De Paola, S. Gaglio, G. Lo Re, F. Milazzo, and M. Ortolani, "Adaptive distributed outlier detection for WSNs," *IEEE Trans. Cybernetics*, vol. 45, no. 5, pp. 888–899, 2015.

[13] D. Hall and S. McMullen, *Mathematical techniques in multisensor data fusion*. Artech House, 2004.

[14] A. De Paola, M. La Cascia, G. Lo Re, M. Morana, and M. Ortolani, "Mimicking biological mechanisms for sensory information fusion," *Biologically Inspired Cognitive Architectures*, vol. 3, pp. 27–38, 2013.

[15] M. C. Huebscher and J. A. McCann, "Adaptive middleware for context-aware applications in smart-homes," in *Proc. 2nd Workshop on Middleware for Pervasive and Ad-Hoc Computing*. ACM, 2004, pp. 111–116.

[16] K. Cho, I. Hwang, S. Kang, B. Kim, J. Lee, S. Lee, S. Park, J. Song, and Y. Rhee, "HiCon: a hierarchical context monitoring and composition framework for next-generation context-aware services," *IEEE Network*, vol. 22, no. 4, pp. 34–42, 2008.

[17] A. Padovitz, S. W. Loke, A. Zaslavsky, B. Burg, and C. Bartolini, "An approach to data fusion for context awareness," in *Modeling and Using Context*. Springer, 2005, pp. 353–367.

[18] N. Roy, S. K. Das, and C. Julien, "Resolving and mediating ambiguous contexts in pervasive environments," *Smart Healthcare Applications and Services: Developments and Practices*, pp. 122–147, 2011.

[19] ——, "Resource-optimized quality-assured ambiguous context mediation framework in pervasive environments," *IEEE Trans. Mobile Computing*, vol. 11, no. 2, pp. 218–229, 2012.

[20] K. P. Murphy, "Dynamic Bayesian networks: representation, inference and learning," Ph.D. dissertation, University of California, Berkeley, 2002.

[21] Y. Zhang and Q. Ji, "Active and dynamic information fusion for multisensor systems with dynamic Bayesian networks," *IEEE Trans. Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 2, pp. 467–472, 2006.

[22] A. De Paola, M. La Cascia, G. Lo Re, M. Morana, and M. Ortolani, "User detection through multi-sensor fusion in an AmI scenario," in *Proc. 15th Int'l Conf. on Information Fusion (FUSION)*. IEEE, 2012, pp. 2502–2509.

[23] A. De Paola, S. Gaglio, G. Lo Re, and M. Ortolani, "Multi-sensor fusion through adaptive Bayesian networks," in *AI*IA 2011: Ar-*

tificial Intelligence Around Man and Beyond. Springer, 2011, pp. 360–371.

[24] E. M. Tapia, S. S. Intille, and K. Larson, *Activity recognition in the home using simple and ubiquitous sensors*. Springer, 2004.

[25] T. van Kasteren and B. Krose, "Bayesian activity recognition in residence for elders," in *Proc. 3rd IET Int'l Conf. on Intelligent Environments (IE)*, 2007, pp. 209–212.

[26] N. Roy, G. Pallapa, and S. K. Das, "A middleware framework for ambiguous context mediation in smart healthcare application," in *Proc. 3rd IEEE Int'l Conf. on Wireless and Mobile Computing, Networking and Communications (WiMOB)*. IEEE, 2007, pp. 72–79.

[27] M. A. Hossain, P. K. Atrey, and A. El Saddik, "Learning multi-sensor confidence using a reward-and-punishment mechanism," *IEEE Trans. Instrumentation and Measurement*, vol. 58, no. 5, pp. 1525–1534, 2009.

[28] G. Fabeck and R. Mathar, "Kernel-based learning of decision fusion in Wireless Sensor Networks," in *Proc. 11th Int'l Conf. on Information Fusion*. IEEE, 2008, pp. 1–7.

[29] D. Cook and S. Das, *Smart environments: technology, protocols and applications*. John Wiley & Sons, 2004, vol. 43.

[30] A. K. Dey, G. D. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-Computer Interaction*, vol. 16, no. 2, pp. 97–166, 2001.

[31] O. Yurur, M. Labrador, and W. Moreno, "Adaptive and energy efficient context representation framework in mobile sensing," *IEEE Trans. Mobile Computing*, vol. 13, no. 8, pp. 1681–1693, 2014.

[32] A. Rahmati and L. Zhong, "Context-based network estimation for energy-efficient ubiquitous wireless connectivity," *IEEE Trans. Mobile Computing*, vol. 10, no. 1, pp. 54–66, 2011.

[33] S. Kang, J. Lee, H. Jang, Y. Lee, S. Park, and J. Song, "A scalable and energy-efficient context monitoring framework for mobile personal sensor networks," *IEEE Trans. Mobile Computing*, vol. 9, no. 5, pp. 686–702, 2010.

[34] S. Nath, "ACE: exploiting correlation for energy-efficient and continuous context sensing," in *Proc. 10th ACM Int'l Conf. on Mobile systems, applications, and services (MobiSys)*. ACM, 2012, pp. 29–42.

[35] Y. Jiang, H. Qiu, M. McCartney, W. G. Halfond, F. Bai, D. Grimm, and R. Govindan, "CARLOG: a platform for flexible and efficient automotive sensing," in *Proc. 12th ACM Conf. on Embedded Networked Sensor Systems (SenSys)*. ACM, 2014, pp. 221–235.

[36] R. J. Meinhold and N. D. Singpurwalla, "Understanding the Kalman filter," *The American Statistician*, vol. 37, no. 2, pp. 123–127, 1983.

[37] D. Sanchez, M. Tentori, and J. Favela, "Hidden Markov Models for activity recognition in Ambient Intelligence environments," in *Proc. 8th Mexican Int'l Conf. on Current Trends in Computer Science*. IEEE, 2007, pp. 33–40.

[38] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT Press, 2005.

[39] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 1, pp. 3–55, 2001.

[40] S. Kullback, R. Leibler *et al.*, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[41] A. De Paola and L. Gagliano, "Design of an adaptive Bayesian system for sensor data fusion," in *Advances onto the Internet of Things*. Springer, 2014, pp. 61–76.

[42] M. Diehl and Y. Y. Haimes, "Influence diagrams with multiple objectives and tradeoff analysis," *IEEE Trans. Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 34, no. 3, pp. 293–304, 2004.

[43] J. A. Tatman and R. D. Shachter, "Dynamic programming and influence diagrams," *IEEE Trans. Systems, Man and Cybernetics*, vol. 20, no. 2, pp. 365–379, 1990.

[44] N. C. Krishnan and D. J. Cook, "Activity recognition on streaming sensor data," *Pervasive and Mobile Computing*, 2012.

[45] L. Gao, A. Bourke, and J. Nelson, "Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems," *Medical Engineering & Physics*, vol. 36, no. 6, pp. 779–785, 2014.

[46] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[47] A. Lalomia, G. Lo Re, and M. Ortolani, "A hybrid framework for soft real-time WSN simulation," in *Proc. 13th IEEE/ACM Int'l Symp. Distributed Simulation and Real Time Applications (DS-RT)*, 2009, pp. 201–207.

Alessandra De Paola received her Bachelor Degree and Master Degree in Computer Engineering from University of Palermo, Italy, in 2004 and 2007, respectively, and Ph.D. in Computer Engineering from University of Palermo, Italy, in 2011. She is an assistant professor of computer engineering at the University of Palermo since 2012. Her current research interests include Artificial Intelligence applied to Distributed Systems, Wireless Sensor Networks, Ambient Intelligence and Network Security.

Pierluca Ferraro is a Ph.D. Student in Computer Engineering at the University of Palermo, Italy. He received his Bachelor Degree and Master Degree in Computer Engineering from the University of Palermo, Italy, in 2010 and 2013, respectively. His current research interests include mobile and pervasive computing, Wireless Sensor Networks, and Ambient Intelligence. He is a student member of the IEEE.

Salvatore Gaglio is full professor of computer science and artificial intelligence at the University of Palermo, Italy, from 1986. From 1998 to 2002 he was the Director of the Study Center on Computer Networks of the CNR (National Research Council); from 2002 he is the Director of the Branch of Palermo of High Performance Computing and Computation of CNR. From 2005 to 2012 he was member of the Scientific Council of the ICT Department of CNR. His present research activities are in the area of artificial intelligence and robotics. He is a member of IEEE, ACM, and AAAI.

Giuseppe Lo Re is associate professor of computer engineering at the University of Palermo. He received the Laurea degree in computer science at the University of Pisa, in 1990 and the Ph.D. in computer engineering at the University of Palermo, in 1999. His current research interests are in the area of computer networks and distributed systems, broadly focusing on Wireless Sensor Networks, Ambient Intelligence, Internet of Things. He is senior member of IEEE and of its Communication Society, and of the Association for Computer Machinery.

Sajal K. Das is the Chair of Computer Science Department and Daniel St. Clair Endowed Chair at the Missouri University of Science and Technology, Rolla. During 2008-2011, he served the US National Science Foundation as a Program Director in the Division of Computer Networks and Systems. His current research interests include wireless sensor networks, mobile and pervasive computing, smart environments and cyber-physical systems, cloud computing, and security. He published extensively with more than 600 research articles in high quality journals and refereed conference proceedings. He holds 5 US patents, coauthored 51 book chapters, and four books. He is a recipient of numerous awards for research, teaching and mentoring. He is the editor-in-chief of Elsevier's Pervasive an Mobile Computing journal and serves on the editorial boards of IEEE TMC and ACM TOSN journals. He is a Fellow of the IEEE.