Appl Intell (2018) 48:966–991
DOI 10.1007/s10489-017-1032-y

Predictive intelligence to the edge through approximate collaborative context reasoning

Christos Anagnostopoulos[1] · **Kostas Kolomvatsos**[2]

Published online: 7 August 2017

Abstract We focus on Internet of Things (IoT) environments where a network of sensing and computing devices are responsible to locally process contextual data, reason and collaboratively infer the appearance of a specific phenomenon (event). Pushing processing and knowledge inference to the edge of the IoT network allows the complexity of the event reasoning process to be distributed into many manageable pieces and to be physically located at the source of the contextual information. This enables a huge amount of rich data streams to be processed in real time that would be prohibitively complex and costly to deliver on a traditional centralized Cloud system. We propose a lightweight, energy-efficient, distributed, adaptive, multiple-context perspective event reasoning model under uncertainty on each IoT device (sensor/actuator). Each device senses and processes context data and infers events based on different local context perspectives: (i) expert knowledge on event representation, (ii) outliers inference, and (iii) deviation from locally predicted context. Such novel approximate reasoning paradigm is achieved through a contextualized, collaborative belief-driven clustering process, where clusters of devices are formed according to their belief on the presence of events. Our distributed and federated intelligence model efficiently identifies any localized abnormality on the contextual data in light of event reasoning through aggregating local degrees of belief, updates, and adjusts its knowledge to contextual data outliers and novelty detection. We provide comprehensive experimental and comparison assessment of our model over real contextual data with other localized and centralized event detection models and show the benefits stemmed from its adoption by achieving up to three orders of magnitude less energy consumption and high quality of inference.

Keywords Collaborative event inference · Federated reasoning · Edge predictive intelligence · Optimal stopping theory · Adaptive vector quantization · Type-2 fuzzy logic inference

1 Introduction

We envisage an IoT environment, where *things* at the edge of the network convey locally inferred knowledge to the IoT applications. We focus on a setting that involves networks of distributed wireless devices (e.g., sensor nodes and actuators, smart meters) capable of sensing and *locally* processing & reasoning about events. Each node performs measurements and locally extracts and infers knowledge over these measurements in light of event reasoning, e.g., wireless sensors spread on a geographical area are responsible for inferring fire or flood incidents. The fundamental requirement to materialize predictive intelligence at the edge of the network is the autonomous nature of nodes to locally perform data sensing & inference, and disseminate **only inferred knowledge** (e.g., minimal sufficient statistics) to their neighbors and *concentrators*. Nodes convey intelligence to concentrators for event inference.

✉ Christos Anagnostopoulos
christos.anagnostopoulos@glasgow.ac.uk

Kostas Kolomvatsos
kolomvatsos@cs.uth.gr

[1] School of Computing Science, University of Glasgow, Glasgow, UK

[2] School of Science, Department of Computer Science, University of Thessaly, Thessaly, Greece

Many critical IoT applications have been developed on top of contextual data streams captured by nodes for events identification and reasoning. Events are related to critical aspects, e.g., security issues or violations of predefined constraints. For instance, in security and environmental monitoring applications, a monitoring infrastructure is imperative to apply an efficient mechanism to derive alerts when specific criteria are satisfied [1, 8, 10, 12, 23]. We can identify two main orientations in terms of data acquisition, transfer and contextual reasoning:

- Orientation 1: Centralized Context Reasoning. Nodes transfer their measurements to a concentrator, e.g., a sink node, back-end-system, Cloud center, which the latter processes data and possesses the intelligence to infer events, and
- Orientation 2: Collaborative Context Reasoning. Nodes locally process data, locally infer knowledge, and have the intelligence for event reasoning in a collaborative manner.

In this paper, we elaborate on the second orientation through a collaborative, intelligent, and adaptive model for local data processing and event reasoning. This *federated reasoning* among nodes involves three *perspectives* of the captured information: (i) predicted context, (ii) contextual inference of outliers, and (iii) context fusion based on expert knowledge. These different Context Perspectives (CPs) are aggregated into a Type-2 Fuzzy Sets inference engine, which locally concludes on an event. Then, through a proposed knowledge-centric nodes clustering scheme in a federated way, nodes disseminate *only* pieces of inferred knowledge among them to unanimously reason about an event based on their local view. In turn, representative nodes of such collaborating clusters locally reason about context and then report the aggregated inference to the concentrators. The concentrators form a *contextual event map* and apply strategies to handle the inferred events, e.g., warn/trigger flood first responders. The key excellence is that our model combines local context processing & inference to the network edge with knowledge-centric nodes clustering. The challenge is to collaboratively process & infer events by minimizing the false alarms / erroneous inference that affect decision making, i.e., unsuitable decisions of handling hazardous phenomena.

1.1 Related work

Event processing & inference is adopted to support the development of IoT applications [8]. From the sensing and processing perspective, normally in the literature, the sensing devices monitor a specific area and deliver the captured data to a back-end system for processing, event inference, and alerts/decision making [10, 24]. Analysis on architectural solutions and case studies on event inference mechanisms is discussed in [3]. The back-end system in [18] adopts aggregate methodologies for event inference, while in [17] it supports IoT applications for air quality monitoring in indoors environments. Such system collects contextual data from temperature, humidity, light, and air quality sensors and then centrally infer events. Moreover, the centralized context reasoning systems in [6, 12], and [23] provide early inference of forest fire events based on vision-enabled sensors, home monitoring based on the received signal strength of sensors, and surveillance of critical areas, respectively. In wireless sensors network deployments, e.g., [2, 5, 11, 16], the back-end systems centrally provide event inference for specific areas by minimizing false alerts.

From the quality of inference perspective, event inference utilizing the principles of approximate reasoning like Fuzzy Logic (FL) is proved a useful technique for delivering high quality of inference. The model in [7] predicts the peak particle velocity of ground vibration levels. Such model adopts a FL-based inference scheme and utilizes the parameters of distance from blast face to the vibration monitoring point. The FL-based context reasoning model in [22] estimates the radiation levels in the air. The adoption of FL aims to handle missing values and, thus, deriving a mechanism capable of delivering alerts. The FL-based fusion model in [35] reduces uncertainty and false-positives within the process of fault detection. In [4], a specific FL-based inference system is proposed for ambient intelligence environments. Such system learns the users' behavior in light of being adapted to the users' profiles. In [13, 14], the authors propose a centralized reasoning system that derives immediate identification of events based only on univariate data. Such system adopts data fusion and prediction for efficiently aggregating sensors measurements. Then, the system adopts FL for handling the uncertainty on the event reasoning.

In all the aforementioned efforts, the edge devices transfer their data to a back-end system, where the latter based on certain computing and reasoning paradigms, e.g., data aggregation, FL-based reasoning, infers events and provides alerts/warnings to IoT applications. The clear major difference of our collaborative machine learning mechanism compared to the aforementioned efforts is the localized event processing & inference at the network edge instead of a centralized reasoning approach. In all research efforts, the back-end system centrally undertakes the responsibility of event reasoning [15] and alerts generation once all contextual data are delivered throughout the network [19].

Our federated reasoning approach drastically departs from the centralized predictive intelligence paradigm to a fully distributed intelligence perspective. Our challenge is to *push the intelligence for event processing & inference* to the edge nodes equipped with computing and sensing

capabilities provide *partial awareness* on an event. By enhancing this local event inference with different CPs, our mechanism (i) avoids raw data transfer from IoT nodes to a back-end system, (ii) favors of conveying the minimal inferred knowledge from the edge to concentrators by introducing a knowledge-centric nodes clustering, (iii) minimizes the false alarm rate by introducing advanced approximate inference over the CPs, and (iv) reduces the communication overhead induced by transferring humongous data volumes from sensors to concentrators through localized inference. In our orientation, the edge nodes do not share and/or relay contextual information. Instead, they conditionally transfer inferred prices of knowledge, if necessary, in light of high quality of inference. Furthermore, from the quality of inference perspective, our mechanism adopts Type-2 Fuzzy Sets over multivariate contextual data instead of univariate data Type-1 Fuzzy Sets as e.g., in [13], to cope with the induced uncertainty of event knowledge representation.

1.2 Research excellence & contribution

To the best of our knowledge, our collaborative machine learning mechanism is a first attempt to materialize the concept of **federated reasoning by conveying predictive intelligence for real-time event inference to the edge of the network**. This is achieved by exploiting at most the computing & sensing capabilities of IoT nodes based on different CPs. *Our vision of intelligent edge computing is materialized by conditionally deliver inferred knowledge from the network edge with high quality of inference* and not transferring data to the back-end-system. In combination with the proposed knowledge-centric clustering scheme, our novel mechanism is robust in terms of erroneous event inference (false alerts) and reduces the communication overhead between nodes and back-end system. The obtained outcome of this research is: (i) accurate event inference close to the source of the contextual information, (ii) significantly low communication overhead by localized belief-centric groupings, thus, avoiding data transfer to the back-end systems, and (iii) *energy-efficient and robust inference* in terms of imprecise and faulty data streams.

The major technical contributions of this research are:

- A temporal nearest-neighbors exponential smoothing model for localized context prediction;
- A conditionally growing adaptive vector quantization model for localized context outliers inference based on the Adaptive Resonance Theory;
- A time-optimized stochastic novelty detection & adaptation model based on the Optimal Stopping Theory. We provide the theoretical analyses for the above-mentioned statistical learning and optimization models;
- A collaborative knowledge-centric nodes clustering scheme and a Type-2 FL-based event inference combining predicted and fused context with outliers identification;
- Asymptotic time and space complexities of the proposed algorithms and collaborative methods and a comprehensive evaluation of the nodes energy consumption in terms of communication and computation/processing cost;
- Performance and comparative assessment of our mechanism with: (i) the *local* voting scheme and (ii) the *centralized* aggregation-based event detection schemes achieving up to three orders of magnitude less energy consumption in an IoT environment.

1.3 Organization

The paper is organized as follows: Section 2 presents the rationale and overview of our federated reasoning approach. Sections 3 and 4 introduce the local context prediction and outliers detection, respectively. Section 5 proposes a novelty & adaptation mechanism, while Sections 6 and 7 introduce context fusion and Type-2 FL-based inference. Section 8 discusses on the collaborative knowledge-centric nodes clustering. Section 9 reports on the asymptotic time and space complexities of the proposed algorithms and methods and discusses the nodes energy consumption in terms of communication and computation/processing cost. Section 10 presents a comprehensive performance and comparative assessment with other event identification mechanisms. Section 11 concludes the paper.

2 Overview & rationale

2.1 Overview

We model the topology of an edge network of sensing and computation nodes (nodes) by an undirected communication graph as shown in Fig. 1 (left). Let $\mathcal{G} = (\mathcal{E}, \mathcal{N})$ denote an undirected graph with vertex set $\mathcal{N} = \{1, 2, ..., n\}$ and edge set $\mathcal{E} \subset \{\{i, j\} | i, j \in \mathcal{N}\}$, where each edge $\{i, j\}$ is an unordered pair of distinct nodes. A graph is connected if for any two vertices i and j there exists a sequence of edges (a path) $\{i, k_1\}, \{k_1, k_2\}, ..., \{k_{s-1}, k_s\}, \{k_s, j\}$ in $\mathcal{E}$. Let $\mathcal{N}_i = \{j \in \mathcal{N} | \{i, j\} \in \mathcal{E}\}$ denote the set of neighbors of node i. Let also a set of concentrator nodes $\mathcal{C} = \{1, \ldots, c\}$ that act as sink nodes for a specific subset of nodes in $\mathcal{N}$. Concentrators gather (digested) context knowledge from certain nodes in order to provide to the IoT applications the corresponding reasoning results by those nodes on the presence of an event of interest. The concentrators could directly connect to a fixed Internet infrastructure, e.g., cloud platform for predictive analytics.

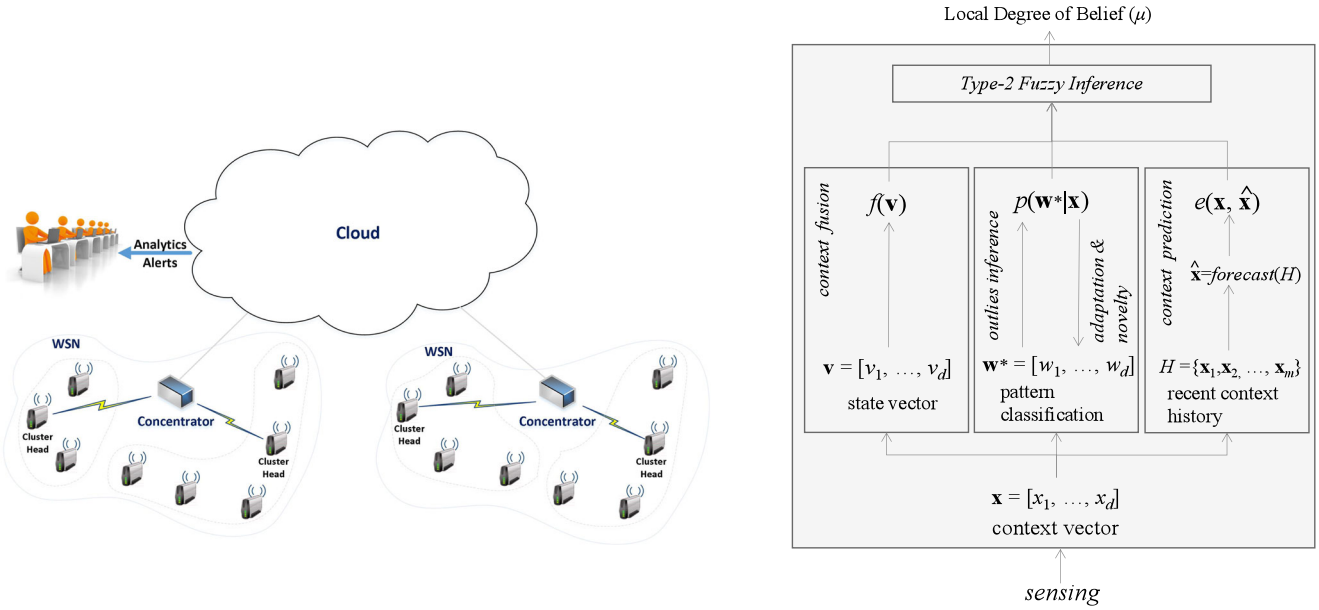

Fig. 1 (*Left*) Overall architecture: IoT nodes locally process data and infer events, where cluster heads (CHs) report the aggregated degree of belief concentrators; (*right*) Internal context processing and reasoning on an IoT node: from context sensing to local event inference

The nodes monitor a specific area by sensing multiple contextual variables like ambient temperature, humidity, wind speed, and perform local reasoning to infer on an event of interest, e.g., a fire or flood event. We assume that nodes observe the same phenomenon. The degree of occurrence or *degree of belief* of an event, notated by μ_i, is locally inferred by node i. This belief is disseminated by node i to its neighbors $\mathcal{N}_i$ to further enhance the contextual knowledge of its neighborhood. This leads to a clustering of nodes according to their view, thus contributing to distributed event reasoning.

The nodes clustering is achieved by the election of a node, referred to as Cluster Head (CH), based only on the disseminated degrees of belief. Groups of nodes are formed each one involving a unique CH. Each CH aggregates its members' degrees of belief and communicates with its concentrator delivering an inference result. In this case, no centralized process is adopted for clustering and data aggregation on event identification. The CHs convey aggregated knowledge to concentrators, thus, minimizing the messages circulated in the network. Note, the messages exchanged among members and CHs are not raw data. Instead, they are pieces of inferred context represented by the degrees of belief as it will be elaborated later. The overall proposed architecture is shown in Fig. 1 (left).

2.2 Rationale

Our multi-perspective collaborative context reasoning model for each node builds on top of a local FL-based inference engine (Type-2 FL System; introduced later) that combines three perspectives of context: (i) *current* fused context, (ii) *predicted* context, and (iii) *outliers* context. This model locally derives the degree of belief μ_i for node i each time a vector of contextual values is captured; hereinafter, referred to as *context vector*. A node i orchestrates the following reasoning processes to infer an event:

- *Context Fusion* evaluates the event inference rule defined by experts from the current context vector.
- *Context Prediction* utilizes the trend of historical context vectors experienced on node i for a short-term forecast of context.
- *Context Outliers & Novelty* incrementally evaluates and revises its belief that the currently context vectors significantly deviate from their statistical patterns experienced on node i.
- *Fuzzy Context Inference*, which is realized by a Type-2 FL System (T2FLS), combines predicted and outliers context vectors with the current fused context. T2FLS derives the μ_i for node i as a local inference.

Assume a discrete time domain $t \in \mathbb{T} = \{1, 2, \ldots\}$. A context (row) vector $\mathbf{x} = [x_1, \ldots, x_d] \in \mathbb{R}^d$ consists of d variables $x_j \in \mathbb{R}$ corresponding to sensor measurements. A node i at time t captures context vector $\mathbf{x}(t)$ and combines the Context Perspectives (CPs):

- (CP1) the current belief of an event by evaluating the expert's knowledge over $\mathbf{x}(t)$.
- (CP2) how much $\mathbf{x}(t)$ is deviated from the predicted context given a short history,

– (CP3) in what degree $\mathbf{x}(t)$ is considered as an *outlier* given the statistical distribution of patterns. Figure 1 (right) shows the all context processing and reasoning processes for node i: from context sensing to local event inference.

Concerning CP1, our model evaluates the belief of event from the current context. Since CP1 constitutes a rule-based baseline solution for event inference, we move a step further to incorporate knowledge from CP2 and CP3. As we show in our evaluation, the fusion of these CPs results to more sophisticated event reasoning.

Concerning CP2, node i stores the most recent m vectors $\mathbf{x}(t-m), \mathbf{x}(t-m+1), \ldots, \mathbf{x}(t-1)$. Based on this history, node i predicts the context vector at time t, $\hat{\mathbf{x}}(t)$ with respect to the conditional expectation conditioned on the recent observed history, i.e.,

$$\hat{\mathbf{x}}(t) = \mathbb{E}[\mathbf{x}(t)|\,\mathbf{x}(t-1), \ldots, \mathbf{x}(t-m)]. \quad (1)$$

Node i then captures the *actual* context $\mathbf{x}(t)$ and the prediction error is $e(t) = \|\mathbf{x}(t) - \hat{\mathbf{x}}(t)\|$, where $\|\cdot\|$ denotes the Euclidean norm. The rationale in CP2 is that the prediction error gives an insight of how the actual vector is deviated from the expected vector based on a short-term history experienced on node i. If the current context deviates from the expected context then this instantaneously indicates that the observed *recent normal* state changes. However, we should take into consideration the statistical patterns from the *entire* history of context vectors to enhance our belief on event inference.

Concerning CP3, node i incrementally estimates the probability distribution of context $p(\mathbf{x})$. This unknown distribution is approximated by specific pattern vectors $\mathbf{w}_k \in \mathbb{R}^d$, $k \in [K]$,[1] which represent the so-far observed vector space $\mathbb{D} \subset \mathbb{R}^d$. The number K of those patterns is not necessarily fixed and is initially unknown. Each pattern $\mathbf{w}_k$ is the representative of the (convex) vector subspace $\mathbb{D}_k \subset \mathbb{D}$. The $p(\mathbf{x})$ is approximated by patterns based on the probability $p(\mathbf{x}|\mathbf{w}_k)$ of observing $\mathbf{x}$ being derived from subspace $\mathbb{D}_k$ represented by $\mathbf{w}_k$. As it will be discussed, this probability depends on the distance between $\mathbf{x}$ and $\mathbf{w}_k$. The rationale in CP3 is that node i infers whether current $\mathbf{x}$ deviates significantly from the (so far) statistical patterns. In turn, node i assesses whether $\mathbf{x}$ lies outside or not the observed vector space utilizing the assignment probability $p(\mathbf{w}^*|\mathbf{x}) \propto p(\mathbf{x}|\mathbf{w}^*)p(\mathbf{w}^*)$, with respect to its closest pattern $\mathbf{w}^*$, i.e.,

$$\mathbf{w}^* = \arg\min_{k\in[K]} \|\mathbf{x} - \mathbf{w}_k\|. \quad (2)$$

As will be discussed, the assignment probability $p(\mathbf{w}^*|\mathbf{x})$ quantifies the instantaneous belief that context is (i) either outlier, (ii) or novelty, thus, expanding our current knowledge, (iii) or a normal instance of the space $\mathbb{D}$, thus, updating our current knowledge. Node i, to support such reasoning, is equipped with a time-optimized mechanism to incrementally update/adjust to possible novel vector subspaces identified, thus, augmenting its current knowledge. This augmentation is achieved by increasing the number of patterns to better reflect the *new* vector subspaces, thus, minimizing the risk of false consideration of outliers, which correspond to false alarms under event inference. Before proceeding with the three CPs, we provide some preliminaries on unsupervised statistical learning and optimal stopping theory adopted in our analysis.

[1] $k \in [K]$ is a compact notation for $k = 1, \ldots, K$ adopted in the paper.

2.3 Preliminaries

2.3.1 Adaptive vector quantization

Adaptive Vector Quantization (AVQ) refers to an unsupervised learning (clustering) algorithm [31] that partitions a d-dimensional space $\mathbb{R}^d$ into a fixed number of K subspaces. AVQ distributes K patterns $\mathbf{w}_1, \ldots, \mathbf{w}_K$ in $\mathbb{R}^d$. A pattern $\mathbf{w}_k$ represents a subspace of $\mathbb{R}^d$. AVQ learns as $\mathbf{w}_k$ changes in response to random vector $\mathbf{x} \in \mathbb{R}^d$. Competition selects which $\mathbf{w}_k$ the vector $\mathbf{x}$ modifies. The k-th pattern 'wins' if $\mathbf{w}_k$ is the closest to $\mathbf{x}$. During partition, vectors $\mathbf{x}$ are projected onto their closest patterns and patterns adaptively move around the space to form optimal partitions (subspaces of $\mathbb{R}^d$) that minimize the *Expected Quantization Error* (EQE):

$$\mathcal{J}(\{\mathbf{w}_k\}) = \mathbb{E}\left[\min_k \|\mathbf{x} - \mathbf{w}_k\|^2\right]. \quad (3)$$

2.3.2 On-line machine learning & stochastic gradient descent

Stochastic Gradient Descent (SGD) [27] is widely adopted in on-line machine learning as an optimization method for incrementally minimizing an objective function $\mathcal{J}(a)$, where $a \in \mathcal{A}$ is a parameter from a parameter space $\mathcal{A}$ and $a^* \in \mathcal{A}$ minimizes $\mathcal{J}$. SGD leads to fast convergence to a^* by adjusting the estimated a so far in the direction (negative gradient $-\nabla\mathcal{J}$), which improves the minimization of $\mathcal{J}$. SGD gradually changes a upon reception of a new training sample. The *standard* gradient descent algorithm updates a as: $\Delta a = -\eta\nabla_a\mathbb{E}[\mathcal{J}(a)]$, where the expectation is approximated by evaluating $\mathcal{J}$ and its gradient over all training pairs and $\eta \in (0, 1)$. On the other hand, SGD simply does away with the expectation in the update of a and computes the gradient of $\mathcal{J}$ using only a single training sample at step $t = 1, 2, \ldots$. The update of a_t at step t is given by:

$$\Delta a_t = -\eta_t \nabla_{a_t}\mathcal{J}(a_t). \quad (4)$$

In SGD, the *learning rate* $\{\eta_t\} \in (0, 1)$ is a step-size schedule, which defines a slowly decreasing sequence of scalars that satisfy:

$$\sum_{t=1}^{\infty} \eta_t = \infty \text{ and } \sum_{t=1}^{\infty} \eta_t^2 < \infty. \tag{5}$$

Choosing the proper learning schedule is not trivial; a practical method is the hyperbolic schedule: $\eta_t = \frac{1}{t+1}$ [27].

2.3.3 Optimal stopping theory

The Optimal Stopping Theory [28] (OST) deals with the problem of choosing the best time instance to take the decision of performing a certain action. This decision is based on sequentially observed random variables in order to maximize the expected reward. For random variables $X_1, X_2, \ldots$ and measurable functions $Y_t = \psi_t(X_1, X_2, \ldots, X_t)$, $t = 1, 2, \ldots$ and $Y_\infty = \psi_\infty(X_1, X_2, \ldots)$, the problem is to find a stopping time τ to maximize $\mathbb{E}[Y_\tau]$. The τ is a random variable with values in $\{1, 2, \ldots\}$ such that the event $\{\tau = t\}$ is in the Borel field (filtration) $\mathbb{F}_t$ generated by $X_1, \ldots, X_t$, i.e., the only available information we have obtained up to t: $\mathbb{F}_t = \mathbb{B}(X_1, \ldots, X_t)$. The decision to stop at t is a function of $X_1, \ldots, X_t$ and does not depend on future observables $X_{t+1}, \ldots$. The problem is to find the optimal stopping time t^* such that the supremum $\mathbb{E}[Y_\tau]$ is attained: i.e.,

$$t^* = \inf\{t \geq 1 | Y_t = \operatorname{ess\,sup}_{\tau \geq t} \mathbb{E}[Y_\tau | \mathbb{F}_t]\}. \tag{6}$$

The (essential) supremum $\operatorname{ess\,sup}_{\tau \geq t} \mathbb{E}[Y_\tau | \mathbb{F}_t]\}$ is taken over all stopping times τ such that $\tau \geq t$. The optimal stopping time t^* is obtained through the *principle of optimality* [30]. The theorem in [39] refers to the existence of the optimal stopping time.

Theorem 1 (Existence of Optimal Stopping Time) *If* $\mathbb{E}[\sup_t Y_t] < \infty$ *and* $\lim_{t \to \infty} \sup_t Y_t \leq Y_\infty$ *almost surely then the stopping time* $t^* = \inf\{t \geq 1 | Y_t = \operatorname{ess\,sup}_{\tau \geq t} \mathbb{E}[Y_\tau | \mathbb{F}_t]\}$ *is optimal.*

Proof See [39]. □

3 Context prediction

The major concept of this CP is to interpret the deviation between the excepted context and the actual context on node i as a *reliable* indication of an event. Context prediction (Fig. 1(right)) involves a multidimensional time-series vector forecast at node i to locally predict the upcoming context $\hat{\mathbf{x}}(t+1)$ given a sliding history window of m observed vectors $\mathbf{x}(t-m), \ldots, \mathbf{x}(t-1)$ and the current context $\mathbf{x}(t)$.

We enhance the multivariate Holt-Winters Double Exponential Smoothing (DES) with a h-Nearest Neighbors smoothing (hNN) at time t, $1 \leq h \leq m$. DES takes into account the possibility of a time series exhibiting some form of trend with an updated slope component. In our case, we attempt to capture the temporal correlation of the noisy contextual data by exploiting the values of the *temporal* data nearest neighbors. The proposed temporal smoothing functionality over DES encapsulates the correlation of values ahead of time, which aligns with our idea of event reasoning using instantaneous context deviation. This deviation should involve the trend and slope, already captured by DES, and the temporal correlation of consequent contextual values. By involving this temporal correlation between recent past and future values, we enhance event reasoning.

Our idea is to substitute each value x_i with the average x_i' of the hNN *backward* and *forward* values, $\forall i$. That is given the values $x_i(k), k = t-h+1, \ldots, t-1$, the corresponding temporal hNN smoothed values $x_i'(k)$ are:

$$x_i'(k) = \frac{1}{h} \sum_{\tau=k-\frac{h-1}{2}}^{k+\frac{h-1}{2}} x_i(\tau) \tag{7}$$

Once x_i' values are smoothed then the forecast of the i-th variable at time t, $x_i(t)$ is achieved using DES, $\forall i$. Evidently, when $h = 1$, then our approach is reduced to DES, i.e., without dealing with the temporal NN smoothing. In turn, we obtain:

$$y_i(t) = \delta x_i'(t) + (1-\delta)(y_i(t-1) + u_i(t-1)) \tag{8}$$

$$u_i(t) = \kappa(y_i(t) - y_i(t-1)) + (1-\kappa)u_i(t-1) \tag{9}$$

where $x_i'(t)$ is the actual smoothed value from our hNN method at t as in (7), $y_i(t)$ and $y_i(t-1)$ are the intercepts at time t and $t-1$, respectively. The $u_i(t)$ and $u_i(t-1)$ are the slopes (time series trends) at time t and $t-1$, respectively. The δ and κ are smoothing constants in (0,1). The δ value is used to smooth the new actual and trend-adjusted previously smoothed intercept, while the κ value is used to smooth the trend. The smoothing constants determine the weight given to most recent past values and control the weight of smoothing. Values close to 1 give weight to more recent values and near to 0 distribute the weights to consider values from the more distant past within the window. We set $\delta = 0.7$ and $\kappa = 0.9$ as in [32].

The expected context vector $\hat{\mathbf{x}}(t) = [\hat{x}_1, \ldots, \hat{x}_d]$ at time t is predicted by the intercept vector $\mathbf{y}(t) = [y_1(t), \ldots, y_d(t)]$ and slope vector $\mathbf{u}(t) = [u_1(t), \ldots, u_d(t)]$ and then we obtain the deviation $e(t) \in [0, 1]$:

$$\hat{\mathbf{x}}(t) = \mathbf{y}(t) + \mathbf{u}(t) \text{ and } e(t) = d^{-\frac{1}{2}} \|\hat{\mathbf{x}}(t) - \mathbf{x}(t)\|, \tag{10}$$

where the factor $d^{-\frac{1}{2}}$ is a normalization factor over the Euclidean norm $\|\cdot\|$ to get a value in [0,1] given that $\mathbf{x} \in [0,1]^d$, i.e., each x_i value is scaled in [0,1].

4 Context outliers inference

This CP infers whether the current context is an outlier, which highly impacts the event reasoning (Fig. 1 (right)). We study the case where outlier context deviates significantly from the up-to-now statistical patterns learned locally on a node. If this deviation occurs regularly, then our model considers the possibility of a novelty, thus, to adapting new knowledge; see Section 5.

4.1 Conditionally growing context vector quantization

Consider a node i, which captures context vectors $\mathbf{x}$ drawn from a space $\mathbb{D}$. Based on those vectors, we identify the vector subspaces $\mathbb{D}_k$, $k \in [K]$, estimate their patterns $\mathbf{w}_k$ and their number K, where $p(\mathbf{x})$ can be approximated. This is achieved by incrementally partitioning the space $\mathbb{D} = \cup_{k=1}^{K} \mathbb{D}_k$. We study an incremental AVQ for partitioning $\mathbb{D}$ into K (unknown) subspaces $\mathbb{D}_k$. The quantization of $\mathbb{D}$ operates as a mechanism to project $\mathbf{x}$ to the closest pattern $\mathbf{w}_k$. Node i incrementally minimizes the EQE:

$$\mathcal{J}(\{\mathbf{w}_k\}) = \mathbb{E}\left[\min_k \|\mathbf{x} - \mathbf{w}_k\|^2\right] \tag{11}$$

We seek the best possible approximation of vectors $\mathbf{x}$ out of a set $\{\mathbf{w}_k\}_{k=1}^{K}$ of (finite) K patterns such that $\mathbf{x}$ is projected to its closest pattern $\mathbf{w}^* \in \mathbb{D}^* \subset \{\mathbf{x} \in \mathbb{D} : \|\mathbf{x} - \mathbf{w}^*\| = \min_k \|\mathbf{x} - \mathbf{w}_k\|\}$. We incrementally minimize $\mathcal{J}$ in (11) with the presence of a random $\mathbf{x}$ and update only the closest pattern $\mathbf{w}^*$. However, the number of subspaces (and, thus, patterns) $K > 0$ is completely unknown and not necessarily constant. The key problem is to decide on an appropriate K value to minimize (11).

In the literature a variety of AVQ methods exists which are not suitable for incremental implementation, because K must be supplied in advance. We propose a *conditionally growing* AVQ algorithm (i) in which the patterns are sequentially updated and (ii) is adaptively growing, i.e., increases K if a criterion holds true. Given that K is not available a-priori, our algorithm minimizes $\mathcal{J}$ with respect to a threshold ρ. Initially, the vector space has a unique (random) pattern, i.e., $K = 1$. Upon the presence of $\mathbf{x}$, our algorithm (i) finds the closest pattern $\mathbf{w}^*$ and (ii) updates $\mathbf{w}^*$ only if the condition $\|\mathbf{q} - \mathbf{w}^*\| \leq \rho$ holds true. Otherwise, $\mathbf{x}$ is currently considered as a *new* pattern, thus, increasing K by one. This conditional quantization leaves random vectors to self-determine the resolution of quantization. Evidently, high ρ would result to coarse space quantization while low ρ yields fine-grained quantization. The parameter ρ is associated with the stability-plasticity dilemma also known as *vigilance* in Adaptive Resonance Theory [29]. In our case, ρ represents a threshold of similarity between vectors and patterns, thus, guiding us in determining whether a new pattern should be formed. To give a physical meaning to ρ, we express it through a set of percentages $a_i \in (0,1)$ of the value ranges of each x_i. Then, $\rho = \|[a_1, \ldots, a_d]\|$ and if we let $a_i = a, \forall i$, then $\rho = (ad)^{1/2}$. High a over high dimensional space results in a low number of patterns and vice versa. The outcome is a set of K patterns $\mathcal{W} = \{\mathbf{w}_k\}_{k=1}^{K}$.

The incremental minimization in (11) given a series of $\mathbf{x}(t), t \in \mathbb{T}$, is achieved by SGD. Our algorithm processes successive $\mathbf{x}(t)$ until a termination criterion $\Gamma(t) \leq \gamma$. $\Gamma(t)$ refers to the distance between successive estimates of the patterns at steps $t-1$ and t. The algorithm stops at the first t where:

$$\Gamma(t) \leq \gamma : \Gamma(t) = \sum_{k=1}^{K} \|\mathbf{w}_k(t) - \mathbf{w}_k(t-1)\|. \tag{12}$$

The update rules of patterns $\mathbf{w}_k$ are provided in Theorem 2.

Theorem 2 *Given context* $\mathbf{x}$ *and its closest pattern* $\mathbf{w}^* \in \mathcal{W}$*, the patterns* $\{\mathbf{w}_k\}_{k=1}^{K}$ *converge to the optimal estimates if updated as:*

$$\Delta\mathbf{w}^* = \begin{cases} \eta(\mathbf{x} - \mathbf{w}^*) & , \text{ if } \|\mathbf{q} - \mathbf{w}^*\| \leq \rho \\ \mathbf{0} & , \text{ otherwise.} \end{cases}$$

Each $\mathbf{w}_k \in \mathcal{W} \setminus \{\mathbf{w}^*\}$ *is updated as:* $\Delta\mathbf{w}_k = \mathbf{0}$*; rate* $\eta \in (0,1)$ *is defined in* Section 2.3.

Proof For proof, see Appendix A.1. □

A fundamental characteristic of our quantization algorithm is that each pattern $\mathbf{w}_k \in \mathcal{W}$ corresponds to the centroid $\mathbb{E}[\mathbf{x}|\mathbf{x} \in \mathbb{D}_k]$ of those vectors $\mathbf{x}$ assigned to $\mathbf{w}_k$. This is utilized for estimating the probability of an outlier as discussed in Section 5.

Theorem 3 *(Centroid Convergence) If* $\bar{\mathbf{x}}$ *is the centroid of the vector subspace* $\mathbb{D}_k$ *and pattern* $\mathbf{w}_k$ *is the closest pattern of those* $\mathbf{x} \in \mathbb{D}_k$*,* $P(\mathbf{w}_k = \bar{\mathbf{x}}) \to 1$ *at equilibrium.*

Proof For proof, see Appendix A.2. □

Our Algorithm 1 processes a (random) context vector one at a time. In the initialization phase, there is only one pattern $\mathbf{w}_1$, i.e., $K = 1$, which is the first vector. For the t-th context $\mathbf{x}(t)$ and onwards, $t \geq 2$, the algorithm: (i) updates the closest pattern to $\mathbf{x}(t)$ (out of K patterns) given that the distance is less than ρ, otherwise (ii) a *new* pattern is added (increasing K by one). The algorithm stops updating the patterns at

the first step t where $\Gamma(t) \leq \gamma$. At that time and onwards, the algorithm returns the set of patterns $\mathcal{W}$ and no further modification is performed.

Algorithm 1 Conditionally growing context vector quantization

Input: vigilance ρ, convergence threshold γ
Result: patterns set $\mathcal{W}$
begin
  Observe context vector $\mathbf{x}$, $\mathbf{w}_1 = \mathbf{x}$, $\mathcal{W} \leftarrow \{\mathbf{w}_1\}$, $K \leftarrow 1$;
  repeat
    Observe next context vector $\mathbf{w}$ and find closest pattern $\mathbf{w}^* = \arg\min_k \|\mathbf{w}_k - \mathbf{x}\|$;
    if $\|\mathbf{w}^* - \mathbf{x}\| \leq \rho$ **then**
      Update $\mathbf{w}^*$ using Theorem 2.
    else
      $K \leftarrow K + 1$, $\mathbf{w}_K \leftarrow \mathbf{x}$, $\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{w}_K\}$;
    end
    Calculate Γ ;
  until $\Gamma \leq \gamma$;
end

4.2 Outliers inference

We study how CP detects a change in the patterns space $\mathbb{D}_k, \forall k$, based only on $\{\mathbf{w}_k\}$ from Section 4.1. Consider an incoming $\mathbf{x}$ to node i. The CP rationale lies in two components: First, decide whether $\mathbf{x}$ is an *outlier* with respect to the current quantization of $\mathbb{D}$. Second, track overtime the number of such outliers and decide that subspaces have changed when this number becomes high.

Consider the probability assignment $p(\mathbf{w}_k|\mathbf{x})$ of $\mathbf{x}$ to a pattern. Since we do not have any prior knowledge about $p(\mathbf{w}_k|\mathbf{x})$, we apply the *principle of maximum entropy*: among all possible probability distributions, we choose the one that maximizes the entropy [34] given an optimal quantization of $\mathbb{D}$. Specifically, $p(\mathbf{w}_k|\mathbf{q})$ conforms to the Gibbs distribution:

$$p(\mathbf{x}|\mathbf{w}_k) \propto \exp(-\beta\|\mathbf{x} - \mathbf{w}_k\|^2), \tag{13}$$

where $\beta \geq 0$ will be explained later. Assuming that each $\mathbf{w}_k$ has the same prior $p(\mathbf{w}_k) = \frac{1}{K}$, through the Bayes' rule $p(\mathbf{w}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{w}_k)p(\mathbf{w}_k)}{p(\mathbf{x})}$ we obtain that:

$$p(\mathbf{w}_k|\mathbf{x}) = \frac{\exp(-\beta\|\mathbf{x} - \mathbf{w}_k\|^2)}{\sum_{i=1}^{K} \exp(-\beta\|\mathbf{x} - \mathbf{w}_i\|^2)}. \tag{14}$$

Note, $p(\mathbf{w}_k|\mathbf{x})$ explicitly depends on the distance of context with patterns. By varying the parameter β, the probability assignment $p(\mathbf{w}_k|\mathbf{x})$ can be completely *fuzzy* ($\beta = 0$, each vector belongs equally to all patterns) and *crisp* ($\beta \to \infty$, each vector belongs to only one pattern, or precisely uniformly distributed over the set of equidistant closest patterns). As $\beta \to \infty$ this probability becomes a delta function around the pattern closest to $\mathbf{x}$. The probability $p(\mathbf{w}^*|\mathbf{x})$ quantifies the belief that $\mathbf{x}$ is an outlier with $\mathbf{w}^*$ being its closest pattern in the quantized space.

5 Context novelty & adaptation

5.1 Context space change detection

The probability assignment $p(\mathbf{w}^*|\mathbf{x})$ is *reconsidered* if $\mathbf{x}$ is *far distant* from $\mathbf{w}^*$. The distance $\|\mathbf{x} - \mathbf{w}^*\|$ quantifies the likelihood that $\mathbf{x}$ is expected to be drawn from $p(\mathbf{x}|\mathbf{w}^*)$ given that $\mathbf{x}$ is assigned to $\mathbf{w}^*$. To decide whether $\mathbf{x}$ can be properly represented by $\mathbf{w}^*$, we associate $\mathbf{w}^*$ with a *dynamic vigilance* $\rho^* > 0$, which depends on the distance of the assigned $\mathbf{x}$ to $\mathbf{w}^*$. This vigilance is a normalized distance ratio of $\|\mathbf{x} - \mathbf{w}^*\|^2$ out of the average distances of all context vectors $\mathbf{x}_\ell$, $\ell = 1, \ldots, L$, that were assigned to $\mathbf{w}^*$:

$$\rho^* = \frac{\|\mathbf{x} - \mathbf{w}^*\|^2}{\frac{1}{L}\sum_{\ell=1}^{L} \|\mathbf{x}_\ell - \mathbf{w}^*\|^2}. \tag{15}$$

Based on this ratio, if ρ^* is less than a threshold $\rho^\top > 0$, $\mathbf{x}$ is properly represented by its closest pattern. Otherwise, $\mathbf{x}$ is deemed to be an *outlier*. A $\rho^\top$ value normally ranges between 2.5 and 5 [33]. Hence, for $\mathbf{x}$, which is assigned to $\mathbf{w}^*$, we define as outlier indicator of $\mathbf{x}$ with respect to $\mathbf{w}^*$ the random variable:

$$I(\mathbf{x}) = \begin{cases} 1 \ , & \text{if } \|\mathbf{x} - \mathbf{w}^*\|^2 > \rho^\top \frac{1}{L}\sum_{\ell=1}^{L} \|\mathbf{x}_\ell - \mathbf{w}^*\|^2 \\ 0 \ , & \text{otherwise.} \end{cases} \tag{16}$$

Let us now move to keeping track of the outlier indicators $I(\mathbf{x}(1)), \ldots, I(\mathbf{x}(t))$ overtime focusing on their closest pattern $\mathbf{w}^*$: $\mathbf{w}^* = \arg\min_k \|\mathbf{x}(t) - \mathbf{w}_k\|$, $\forall t$. To simplify the notation, we set $I_t = I(\mathbf{x}(t))$. A cumulative sum of I_t's with a high portion of 1's causes node i to consider that $p(\mathbf{x}|\mathbf{w}^*)$ might have changed. Upon observation of $\mathbf{x}$, node i observes for pattern $\mathbf{w}^*$ the random variables $\{I_1, \ldots, I_t\}$. Node i detects a change in $p(\mathbf{x}|\mathbf{w}^*)$ based on the cumulative sum S_t of the $I_1, I_2, \ldots, I_t$ up to t-th assigned vector:

$$S_t = \sum_{\tau=1}^{t} I_\tau. \tag{17}$$

I_t is a discrete random process with independent and identically distributed (i.i.d.) samples. Each I_t follows an unknown probability distribution depending on the distance of $\mathbf{x}$ to $\mathbf{w}^*$. I_t has finite mean $\mathbb{E}[I_t] < \infty$, $t = 1, \ldots$, which

depends on $\|\mathbf{x}(t) - \mathbf{w}^*\|^2$ and the expectation of an outlier indicator is:

$$\mathbb{E}[I] = 0 \cdot P(\{I = 0\}) + 1 \cdot P(\{I = 1\}) = P(\{I = 1\}) \quad (18)$$

Our knowledge on that distribution, which is not trivial to estimate, will provide insight to judge whether $p(\mathbf{x}|\mathbf{w}^*)$ has changed in the subspace determined by $\mathbf{w}^*$. We should 'follow' the trend of that change by either updating $\mathbf{w}^*$, to continuously represent its subspace or, create a new pattern in the novel vector subspace.

By observing I_t and sum S_t up to t, the challenge here is to decide how large the sum should get before deciding that $p(\mathbf{x}|\mathbf{w}^*)$ has changed. Should we decide at an early stage that $p(\mathbf{x}|\mathbf{w}^*)$ has changed, this might correspond to 'premature' decision; a relatively small number of 'outliers' might not correspond to change in $p(\mathbf{x}|\mathbf{w}^*)$. Should we 'delay' our decision then we might get erroneous event inference (high false alarm rate), since we avoid adapting $\mathbf{w}^*$ to 'follow' the trend of the vector subspace change.

The rationale for this CP has as follows: To decide when $p(\mathbf{x}|\mathbf{w}^*)$ has changed we could wait for an *unknown* finite horizon t^* in order to be more confident on a change. During the t^* horizon, we only observe the cumulative sum $S_\tau, \tau = 1, \ldots, t^*$. We propose a stochastic optimization algorithm that postpones a vector space change decision through additional observations of I_τ. At time t^*, a decision on a possible $p(\mathbf{x}|\mathbf{w}^*)$ change has to be taken. The problem is to find the optimal stopping time t_* in order to ensure that $p(\mathbf{x}|\mathbf{w}^*)$ has changed from those $\mathbf{x}(t)$ assigned to $\mathbf{w}^*$ at $t > t^*$.

We define our *confidence* Y_t of a decision on a change of $p(\mathbf{x}|\mathbf{w}^*)$ based on the cumulative sum S_t in (17). Y_t is directly connected to the performance improvements that a timely decision yields. Y_t is a random variable generated by the sum of I_τ up to t, $S_t = \sum_{\tau=1}^{t} I_\tau$, discounted by a risk factor $\alpha \in (0, 1)$:

$$Y_t = \alpha^t S_t. \quad (19)$$

Our algorithm has to find t^* in order to (i) either start adapting $\mathbf{w}^*$ after considering that $p(\mathbf{x}|\mathbf{w}^*)$ has changed or (ii) create a *new* pattern, with respect to vigilance ρ (see Section 4) for those vectors arrive at $t > t_*$. If we never start this adaptation, our confidence that we follow the *new* trend (patterns) is zero, $Y_\infty = 0$. This indicates that we do not 'follow' the trend of a possible change over the subspace and/or do not augment further our knowledge on possibly new vector subspaces. Furthermore, we will never start adapting $\mathbf{w}^*$ at some t with $S_t = 0$, since there is no piece of evidence of any outlier up to t. As I_t assumes unity values for certain times then S_t increases at a high rate, thus indicating a possible change due to a significant number of outliers. Our problem is to decide how large the S_t should get before we start adapting $\mathbf{w}^*$ or augment our current knowledge on the underlying vector space distribution by adding extra patterns. We have to find a time $t > 0$ that maximizes our confidence, i.e., when the supremum

$$\sup_t \mathbb{E}[Y_t] \quad (20)$$

is attained. The semantic of the risk factor α has as follows. High α indicates a conservative adaptation model; it requires additional observations for concluding on a change decision. This, however, comes at the expense of possible outliers prediction inaccuracies during this period, since the $\mathbf{w}^*$ might not be a representative of its corresponding assigned vectors. Low α denotes a rather optimistic model, which reaches premature decisions on a $p(\mathbf{x}|\mathbf{w}^*)$ change. This means that once we concluded on a change, we have to adapt $\mathbf{w}^*$ by actually exploiting every incoming vector assigned to $\mathbf{w}^*$ and/or considering $\mathbf{x}$ as a new pattern. This continues until the updated $\mathbf{w}^*$ converges.

We propose a solution for the problem in (20). Firstly, we prove the existence of t_* in our case, then report on the corresponding optimal stopping time, and finally elaborate on the optimality of the proposed solution. A decision taken at time t is:

- either to assert that a change on $p(\mathbf{x}|\mathbf{w}^*)$ holds true and, then, start the adaptation of $\mathbf{w}^*$ or inserting $\mathbf{x}$ as a new pattern,
- or continue the observation process at time $t + 1$ and, then, proceed with a decision.

Based only on $S_t = \sum_{\tau=1}^{t} I_\tau$ we determine a stopping time that maximizes (20).

Theorem 4 *An optimal stopping time t^* for the problem in (20) exists.*

Proof For proof, see Appendix A.3. □

In our case, I_t are non-negative, thus, the problem is *monotone* [28]. This means that t^*, since it exists by Theorem 4, is obtained by the *1-stage look-ahead optimal rule* (1–sla) [28]. That is, we should start adapting $\mathbf{w}^*$ at the *first* stopping time t at which $Y_t \geq \mathbb{E}[Y_{t+1}|\mathbb{F}_t]$, i.e.,

$$t^* = \inf\{t \geq 1 | Y_t \geq \mathbb{E}[Y_{t+1}|\mathbb{F}_t]\}. \quad (21)$$

For our monotone stopping problem with observations $I_1, I_2, \ldots$ and rewards $Y_1, Y_2, \ldots, Y_\infty$, the 1–sla is optimal since $\sup_t Y_t$ has finite expectation ($\mathbb{E}[I]\frac{\alpha}{1-\alpha}$) and $\lim_{t\to\infty} \sup_t Y_t = Y_\infty = 0$ (see Theorem 4).

Theorem 5 *The optimal stopping time t^* for the problem in (20) is $t^* = \inf\{t \geq 1 | S_t \geq \frac{\alpha}{1-\alpha}\mathbb{E}[I]\}$.*

Proof For proof, see Appendix A.4. □

To derive t^* from Theorem 5 we need to estimate the expectation $\mathbb{E}[I] = P(\{I = 1\})$. Empirically, the probability $P(\{I = 1\})$ can be experimentally calculated by those assigned vectors whose ratio of the distances from their closest patterns out of the total variance of the distances is at least ρ; refer to (15). Moreover, we provide an estimate for $P(\{I = 1\})$ based on our quantization algorithm in Section 4. The probability of $\{I_t = 1\}$ refers to the conditional probability of $\mathbf{x}(t)$ being an outlier given that it is assigned to $\mathbf{w}^*$ with $p(\mathbf{w}_*|\mathbf{x}(t))$. The $P(\{I_t = 1\})$ is, therefore, associated with the probability that the distance $\|\mathbf{x}(t) - \mathbf{w}^*\|^2 > \theta$, with scalar:

$$\theta = \rho^\top \frac{1}{L} \sum_{\ell=1}^{L} \|\mathbf{x}_\ell - \mathbf{w}^*\|^2. \tag{22}$$

If we define the vector $\mathbf{z}(t) = \mathbf{x}(t) - \mathbf{w}^*$ then we seek the probability density distribution of its squared Euclidean norm $\|\mathbf{z}(t)\|^2$. Therefore, based on the centroid convergence in Theorem 3, $\mathbf{w}^*$ refers to the centroid: $\mathbf{w}^* = \mathbb{E}[\mathbf{x}|\mathbf{x} \in \mathbb{D}^*]$. Hence, the squared distance of $\mathbf{z} = [z_1, \ldots, z_d] = [x_1 - w_1^*, \ldots, x_d - w_d^*]$ under the assumption of normally distributed random components follows a non-central squared χ^2 distribution $\chi^2(d, \zeta)$ with d degrees of freedom and non-centrality parameter $\zeta = \sum_{i=1}^{d} (w_i^*)^2$. We approximate $P(\{I = 1\}) = P(\|\mathbf{z}\|^2 > \theta) = 1 - P(\|\mathbf{z}\|^2 \leq \theta)$ by the cumulative distribution function $CDF_{\chi^2(d,\zeta)}(\theta) = P(\|\mathbf{z}\|^2 \leq \theta)$ of $\chi^2(d, \zeta)$. Let $Q_{\kappa_1}(\kappa_2, \kappa_3)$ be the monotonic, log-concave Marcum Q-function, with parameters κ_1, κ_2, and κ_3. Then, we obtain that $CDF_{\chi^2(d,\zeta)}(\theta) = P(\|\mathbf{z}\|^2 \leq \theta) = 1 - Q_{\kappa_1}(\kappa_2, \kappa_3)$:

$$P(\{I = 1\}) = 1 - CDF_{\chi^2(d,\zeta)}(\theta) = Q_{\frac{d}{2}}\left(\sqrt{\zeta}, \sqrt{\theta}\right) \tag{23}$$

by substitution in the Q function: $\kappa_1 = \frac{d}{2}$, $\kappa_2 = \sqrt{\zeta}$, and $\kappa_3 = \sqrt{\theta}$. For an analytical expression of (23), refer to Appendix A.7. Hence, the optimal stopping time is obtained once we substitute $\mathbb{E}[I]$ in Theorem 5 by the $P(\{I = 1\})$ estimated in (23).

5.2 Context adaptation

Once node i has detected a change in at least one vector subspace then it initiates a process that adapts the patterns by modifying $\mathbf{w}^*$ as follows. A change in a vector subspace indicates that new patterns can be formed or existing patterns should be updated. Node i for every incoming $\mathbf{x}$ appearing at $t > t^*$ updates either $\mathbf{w}^*$ to follow the trend or create a new pattern $\mathbf{w}_{K+1} = \mathbf{x}$ as described in Algorithm 1. Algorithm 2 shows the change detection and adaptation process.

Algorithm 2 Context change detection & adaptation

Input: Risk factor $\alpha \in (0, 1)$, patterns set $\mathcal{W}$
Output: Updated patterns set $\mathcal{W}$
Calculate $\theta_k \leftarrow \rho^\top \frac{1}{L_k} \sum_{\ell=1}^{L_k} \|\mathbf{x}_\ell - \mathbf{w}_k\|^2$, $\mathbb{E}_k[I]$ using (23), $t \leftarrow 0$, $S_0^k \leftarrow 0$, $\forall k \in [K]$;
begin
  /* optimal change detection time by incremental outliers indicators */
  repeat
    Observe context $\mathbf{x}(t)$ and assigns $\mathbf{x}(t)$ to its closest pattern $\mathbf{w}_k$;
    Calculate the outlier indicator I_t for $\mathbf{w}_k$ using (16);
    $S_t^k \leftarrow S_t^k + I_t^k$, $t \leftarrow t + 1$;
  until $S_t^k \geq \frac{\alpha}{1-\alpha} \mathbb{E}_k[I]$;
  /* adaptation: either updating closest pattern or expand patterns set */
  if $\|\mathbf{w}_k - \mathbf{x}\| \leq \rho$ **then**
    Update $\mathbf{w}_k$ using Theorem 2.
  else
    $K \leftarrow K + 1$, $\mathbf{w}_K \leftarrow \mathbf{x}$, $\mathcal{W} \leftarrow \mathcal{W} \cup \{\mathbf{w}_K\}$;
  end
end

6 Expert knowledge context fusion

This CP evaluates the belief of an event based on experts' knowledge (Fig. 1 (right)). Consider the context $\mathbf{x}$ at node i. Each variable x_j, $j = 1, \ldots, d$ in $\mathbf{x}$ affects the event reasoning in a different way, as interpreted by human expert knowledge. For instance, consider the identification of a fire event. A fire event can be inferred based on temperature x_1, humidity x_2, and (ionization) smoke x_3 measurements, i.e., $\mathbf{x} = [x_1, x_2, x_3]$. A human expert can express a fire event through an increment on temperature and smoke, with humidity remaining at relatively low levels. Let the row vector $\mathbf{x}_P$ be constructed by variables from $\mathbf{x}$ that proportionally affect the presence of an event, i.e., the event is expressed by an increment on the values for those variables. Similarly, let the row vector $\mathbf{x}_N$ be constructed by the variables from $\mathbf{x}$ that do not proportionally affect the presence of the event, i.e., the event is expressed by a decrease on the values of those variables. In this case, we obtain $\mathbf{x} = [\mathbf{x}_P; \mathbf{x}_N]$, where in our example we have that $\mathbf{x}_P = [x_1, x_3]$ and $\mathbf{x}_N = [x_2]$. This classification of the x_j variables into the $\mathbf{x}_P$ and $\mathbf{x}_N$ vectors is provided directly by the human

interpretation of an event. Based on this representation, we introduce a vector fusion function that produces a unified view on the event identification. We introduce the normalized 'state' $v_j \in [0, 1]$ of each x_j from $\mathbf{x}_P$ and $\mathbf{x}_N$:

$$v_j = \begin{cases} \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}}, & x_j \in \mathbf{x}_P \\ \frac{x_j^{\max} - x_j}{x_j^{\max} - x_j^{\min}}, & x_j \in \mathbf{x}_N \end{cases} \tag{24}$$

The state v_j indicates whether $x_j \in \mathbf{x}_P$ (or $\in \mathbf{x}_N$) has reached its maximum (or minimum) value and, thus, it *partially* expresses the existence of an event. Define $\mathbf{v} = [v_1, \ldots, v_d] \in [0, 1]^d$, which contains the states of all variables from $\mathbf{x}$. Motivated by the sigmoid function from neural computation for activating the impact of each neuron input (the states variables in our case), we adopt the sigmoid product fusion function $f : [0, 1]^d \to \mathbb{R}^+$, which returns the entire state of vector $\mathbf{x}$, i.e., the existence of an event if $f(\mathbf{v}) \to 1$, or not if $f(\mathbf{v}) \to 0$, with:

$$f(\mathbf{v}) = \prod_{j=1}^{d} \frac{1}{1 + \exp\left(-\lambda_2 v_j + \lambda_1\right)}. \tag{25}$$

Function f fuses the current context vector into a scalar indicating the presence of an event through the normalized states v_j. The $\lambda_1, \lambda_2 \in \mathbb{R}$ parameters are application specific. Through the adopted sigmoid function, we can either eliminate or pay more attention on the value of a given variable x_j to the fusion result. For instance, we count a high impact of v_j when its value is only above threshold λ_1 by setting $\lambda_2 \to 0$ (tuning the steepness of the sigmoid function).

7 Event inference under uncertainty

7.1 Fuzzy contextual knowledge base

Based on the CPs in Sections 3, 4, and 6, node i locally achieves event inference at time t by considering (i) the current context fusion $f(\mathbf{v}(t))$ in (25), (ii) the current assignment probability $p(\mathbf{w}^*|\mathbf{x}(t))$ in (14) w.r.t. to closest pattern $\mathbf{w}^*$, and (iii) the current deviation $e(t)$ in (10) for $\mathbf{x}(t)$. We attempt to fuse these CPs through a finite set of Fuzzy Inference Rules (FIR). Each FIR reflects the degree of belief for a specific event inferred locally on node i. For instance, a FIR is: 'when the local sensed temperature is high then the degree of belief for a fire event might be also high'. We propose a T2FLS, which defines the fuzzy knowledge base of FIRs for node i. In this work, we do not rely on a Type-1 FLS (T1FLS) as such an inference model has specific drawbacks when applied in dynamic environments and, more interestingly, when the construction of the FIRs involves uncertainty due to partial knowledge in representing the output of the inference result [21]. In our case, this corresponds to the uncertainty of defining the occurrence of an event based only on the local available knowledge: current context, predicted context, and possible outliers. The limitation in a T1FLS is on handling uncertainty in representing knowledge through FIRs [9, 21]. In a T1FLS, the experts define exactly the membership degree of the involved input and output variables in a FIR, e.g., the characterization of a value as 'high' or 'low'. However, when even the definition of a membership function involves uncertainty, the experts cannot be certain about the membership grade. In such cases, uncertainty is observed not only on the environment of the examined problem, e.g., we classify a value as 'high' or 'low' or the degree of belief as 'high', but also on the description of the term e.g., 'high', itself in a FIR.

In a T2FLS, the membership functions that characterize the terms of the three CPs are themselves 'fuzzy', which leads to the definition of FIRs incorporating such uncertainty [21]. This approach seems appropriate in our case as FIRs cannot explicitly reflect knowledge on whether incoming measurements correspond to the occurrence of an event. Our FIRs take into consideration the uncertainty in the definition of an event by the human expert enhanced with the CPs: deviation of predicted context and outliers inference. Such FIRs refer to a non-linear mapping $\mathcal{F}(f(\mathbf{v}), p(\mathbf{w}^*|\mathbf{x}), e)$ between the three CPs (inputs) and one output, i.e., the degree of belief $\mu_i \in [0, 1]$. The antecedent part of a FIR is a linguistic conjunction of the CPs and the consequent part is the degree of belief that event *actually* occurs. The structure for a FIR is as follows:

IF $f(\mathbf{v})$ is A_{1k} **AND** e is A_{2k} **AND** $p(\mathbf{w}^*|\mathbf{x})$ is A_{3k}

THEN μ_i is B_k,

where A_{1k}, A_{2k}, A_{3k} and B_k are membership functions for the k-th FIR mapping the values of $f(\mathbf{v})$, e, $p(\mathbf{w}^*|\mathbf{x})$ and μ_i into unity intervals, respectively, by *characterizing* these values through the linguistic terms: *low*, *medium*, and *high*. If a linguistic term, e.g., 'high', was represented through one fuzzy set in a T1FLS then we would use one membership function $g(x) \in [0, 1]$ mapping the real value (input) $x \in [0, 1]$ to a discrete set of pairs $(x_j, g(x_j))$, e.g., $\{(0, 0); (0.25, 0.1); (0.5, 0.75); (1, 1)\}$, where $(0.25, 0.1)$ means that the value $x = 0.25$ has a membership degree $g(x) = 0.1$.

In a T2FLS, each term A_{1k}, A_{2k}, A_{3k} and B_k in FIRs is represented by two membership functions corresponding to lower and upper bounds [20]. For instance, the term 'high', unlike in a T1FLS, whose membership for each x is a number $g(x)$, is represented by two membership functions. That is, each value x is assigned to an interval $[g_L(x), g_U(x)]$ corresponding to a lower and an upper membership function g_L and g_U, respectively. E.g., the

membership of $x = 0.25$ is the interval $[0.05, 0.2]$. The interval areas $[g_L(x_j), g_U(x_j)]$ for each input x_j reflect the uncertainty in defining the term, e.g., 'high', which is useful when it is difficult to determine the exact membership function for each term or in modeling the diverse opinions from different CPs in defining the occurrence of an event, in our case. If $g_L(x) = g_U(x), \forall x$, we obtain a FIR in a T1FLS. Following the above FIR structure, each A_{jk}, $j = 1, 2, 3$, and B_k, for each k-th FIR, corresponds to a set of intervals. The interested reader could also refer to [20] for fuzzy reasoning in T2FLS.

7.2 Determination of local degree of belief

A μ_i value close to unity denotes the case where the belief is at high levels, i.e., there is a high belief that a hazardous phenomenon, like fire or flood, occurs in the area of interest based on the *agreement* of the three CPs (all of them assume values close to unity). The opposite stands when μ_i tends to zero. We consider three fuzzy linguistic terms for the FIRs: *Low*, *Medium*, and *High*. *Low* represents that a variable (input or output) takes values close to 0, while *High* depicts the case where a variable takes values close to 1. *Medium* depicts the case where the variable takes values around 0.5. For instance, a *Low* fuzzy value for e indicates that the current and predicted context are close enough, thus, current context *follows* the trend of its recent historical context. A *High* fuzzy value for $p(\mathbf{w}^*|\mathbf{x})$ denotes that the current context does not significantly deviate from its regular statistical pattern. A *High* fuzzy value for $f(\mathbf{v})$ indicates a positive inference on the presence of an event as represented by an expert's knowledge. For each fuzzy term, human experts define the upper and the lower membership functions. Here, we consider triangular membership functions g_L and g_U as they are widely adopted in the literature. Our T2FLS is generic, thus, any type of membership functions can be adopted to better suit to the application domain.

Table 1 shows the proposed fuzzy knowledge based for event inference.[2] Upon receiving the current context $\mathbf{x}(t)$, node i produces its corresponding (i) fused context $f(\mathbf{v})$, (ii) deviation $e(t)$ and (iii) assignment probability $p(\mathbf{w}^*|\mathbf{x})$. Then, the T2FLS is activated as follows: (Step 1) calculation of the interval (based on the membership functions) for each input; (Step 2) calculation of the active interval of each FIR; (Step 3) performance of 'type reduction' to combine the active interval of each FIR and the corresponding consequent. Step 3 produces the interval of the consequent, and accordingly, the defuzzification phase[3] determines a scalar value for the local degree of belief μ_i at time t. The most common method for 'type reduction' is the *center of sets type reducer* [21], which generates a Type-1 Fuzzy Set as output, which is then converted in a scalar value for the μ_i after defuzzification. When the μ_i is over a pre-defined belief threshold $\epsilon \in [0, 1]$, the T2FLS engine infers locally an event occurrence with degree of belief μ_i.

[2] 'Any' in FIRs refers to fuzzy values: 'Low', 'Medium', 'High'.

[3] Defuzzification is the process of producing a quantifiable result in FL, given fuzzy sets and corresponding membership degrees.

8 Belief-centric clustering

In our federated reasoning approach, groups of nodes are formated based on their local degrees of belief μ_i, $i \in \mathcal{N}$. The clustering process is repeated at a *clustering era* $T_1, T_2, T_3, \ldots, T_n \in \mathbb{T}$. The T_n is a variable time index in $\mathbb{T}$, which is triggered by node i which *locally* believes in an event presence in the first instance (i.e., $\mu_i \geq \epsilon$), thus, asking for the opinions of its local neighbors *before* reaching a conclusion. In each group, a node is elected as the Cluster Head (CH) and is responsible to exchange the aggregated degrees of belief (discussed later) with a concentrator from set $\mathcal{C}$ after a belief revision/update of the initial opinion on an event presence. Hence, the number of messages circulated in the network is reduced as it is not necessary for each node to relay messages to a concentrator. The election process concerns a node i to become a CH if it experiences the highest μ_i related to an observed phenomenon among its neighbors $\mathcal{N}_i$. The aim of the CH is to notify its members about its appointment as a CH, thus, avoiding redundant message dissemination. The CH node, after its appointment, aggregates the degrees of belief of its neighbors resulting to an enhanced neighborhood contextual knowledge by unanimously inferring a possible event.

The primary objectives of the federated election process are:

- (i) Appointment of a subset of nodes as CHs responsible for determining and disseminating an unanimous (aggregated) degree of belief to the concentrators.
- (ii) Dynamically changing the CH appointment to nodes. Evidently, this prolongs the network lifetime by changing CH appointments and, thus, balancing energy consumption for the event inference process and transmission of message to the members and concentrators.
- (iii) Termination of the election process within a constant number of iterations (exchanged messages).

It should be noted that the description of the CH replacement process (i.e., objective (ii)) is beyond the scope of this paper. It is also worth noting that we do not make any assumption about the spatial distribution of IoT nodes in the area. Every node can act as either CH or member. This requires the need for an efficient CH election algorithm.

Table 1 T2FLS fuzzy knowledge base

	Context perspectives			Local degree of belief
FIR	$f(\mathbf{v})$	e	$p(\mathbf{w}^*\|\mathbf{x})$	μ_i
1	Low	Any	Low	Medium
2	Low	Any	Medium or High	Low
3	Medium	Any	Any	Medium
4	High	Low or Medium	Low or Medium	High
5	High	Low	High	High
6	High	Medium	High	Medium
7	High	High	Any	Medium

8.1 Belief-centric cluster-head election

A baseline solution for the election process involves nodes exchanging their μ_i to all neighbors. The node with the highest μ_i is elected to become the CH of the neighborhood. However, this solution requires a significant number of messages exchanged among nodes. Moreover, since the election process is re-initiated after a time interval T, then a high energy budget is required for that type of communication. There are certain election algorithms which could be adopted. In our case, neighboring nodes exchange their μ_i values and then 'elect' the CH. To this end, we follow the concept of the CH election algorithm in [37] by modifying the election criteria to reflect the knowledge exchange over a neighborhood.

At each node, the election process requires a number of iterations $L > 0$. In every iteration, nodes send and receive specific small-sized messages from neighbors containing their degrees of belief. Before node i starts the election process, it configures a local probability of becoming a CH ξ_i, hereinafter referred to as Election Probability (EP), as a function of μ_i, i.e., $\xi_i = \max(\xi_{\min}, \mu_i)$, where ξ_{min} is a minimum EP for each node: ξ_i is not allowed to fall below the $\xi_{\min}$, e.g., 10^{-3}. This restriction is essential for terminating the election process in $L = O(1)$ iterations; see Lemma 1. Node i with a high EP ξ_i starts the following process: it sends announcement messages of the form $\langle \xi_i, i \rangle$ to the $\mathcal{N}_i$ neighbors to be a CH. A node j with a low EP ξ_j delays the transmission of announcement messages and considers itself 'non-CH' if it has heard from $\langle \xi_i, i \rangle$ with $\xi_i > \xi_j$. During iteration ℓ, $1 \leq \ell \leq L$, every node i decides to become a CH with EP ξ_i. Through the process, node i can either be elected to become a CH according to its EP ξ_i or remain at the same status (i.e., non-CH) according to overheard announcement messages within its neighborhood $\mathcal{N}_i$. A node j selects its CH node i to be the node with the highest μ_i; this is achieved by the comparison of ξ_i and ξ_j. Every node i then multiplies its EP ξ_i with a factor of $\chi > 1$, and goes to the next step $\ell + 1$ and so on, i.e.,

$$\xi_i(\ell + 1) = \min(\chi \xi_i(\ell), 1). \tag{26}$$

If node i decides to become a CH since its EP ξ_i has reached unity, it sends an announcement message 'CH i' to its neighbors $\mathcal{N}_i$. A node $j \in \mathcal{N}_i$, then, considers itself 'non-CH' if it has heard from node i a 'CH i' message and terminates the election process. Note, this election process is completely distributed. Node i either decides to become a CH since μ_i is the highest among its neighbors, or be a member which awaits a message by its unique CH.

Lemma 1 *The belief-centric election process requires $O(1)$ iterations.*

Proof For proof, see Appendix A.5. □

The number of iterations for each node does not depend on the number of neighbors and is bounded by a constant. Indicatively, when $\xi_{min} = 10^{-3}$ and $\chi = e$ then a node needs at most eight iterations to elect or be elected as a CH.

Lemma 2 *The message exchange complexity in the belief-centric election process is $O(1)$ per node and $O(|\mathcal{N}|)$ for the network.*

Proof For proof, see Appendix A.6. □

8.2 Aggregated degree of belief & federate event reasoning

Once node i is appointed as a CH, it locally determines the average degree of belief of its members $j \in \mathcal{N}_i$:

$$\bar{\mu}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mu_j. \tag{27}$$

The $\bar{\mu}_i$ reflects a degree of consensus of the neighborhood on event inference. CH i, based on the pair $(\mu_i, \bar{\mu}_i)$, determines an *aggregated degree of belief* $\tilde{\mu}_i$. We adopt a reward-idle methodology to reason on the aggregated degree of belief $\tilde{\mu}_i$, which will be delivered by CH i to its

concentrator. If CH i and its neighbors unanimously agree on the presence of an event, i.e., if the logical expression:

$$(\mu_i \geq \epsilon) \wedge (\bar{\mu}_i \geq \epsilon) \tag{28}$$

holds true then we reward CH i's belief on the event by sending to the concentrator $\tilde{\mu}_i = \mu_i$. When CH i and its neighbors unanimously agree on the absence of an event, i.e., if it holds true that:

$$(\mu_i < \epsilon) \wedge (\bar{\mu}_i < \epsilon) \tag{29}$$

then $\tilde{\mu}_i$ is the average value of all degrees of belief:

$$\tilde{\mu}_i = \frac{1}{|\mathcal{N}_i| + 1} \left(\sum_{j \in \mathcal{N}_i} \mu_j + \mu_i \right), \tag{30}$$

and the CH i does not notify the concentrator. If there is a disagreement between CH i and its neighborhood, i.e., if it holds true that:

$$(\mu_i > \epsilon) \wedge (\bar{\mu}_i < \epsilon) \tag{31}$$

then CH i notifies its concentrator after regulating its local opinion by a factor of $r \in (0, 1)$ towards the neighbors' average belief, i.e.,

$$\tilde{\mu}_i = \mu_i + r(\bar{\mu}_i - \mu_i). \tag{32}$$

The concentrator then acquires knowledge for a specific region of the area of interest about the appearance of an event and to what extend this local inference from nodes $\{i, \mathcal{N}_i\}$ is of high belief by receiving $\tilde{\mu}_i$. Note, since $\mu_i \geq \max_{j \in \mathcal{N}_i}\{\mu_j\}$, there will be never the case: $(\mu_i < \epsilon) \wedge (\bar{\mu}_i > \epsilon)$.

9 Computational complexity, energy & communication cost

In this section we present the time and space computational complexities for both Algorithms 1 and 2 and the energy and communication cost of the processes for each node i: (i) event inference (local derivation of degree of belief μ_i, (ii) election and clustering era (appointment of cluster-heads CH and cluster members), (iii) derivation of aggregated degree of belief from the CHs $\tilde{\mu}_i$, and (iv) report to the concentrators from CHs.

9.1 Computational complexity

We report on the time and space complexities of the processes that are needed for each node i to locally infer the degree of belief. such processes include: (i) context vector quantization for patterns derivation, (ii) change detection of the quantized data subspace, (iii) context adaptation, and (iv) degree of belief inference including context prediction and fuzzy inference.

9.1.1 Time & space complexity for context vector quantization

The Algorithm 1 is an incremental partitioning algorithm which updates its closest current pattern $\mathbf{w}^*$ based on the incoming context vector $\mathbf{x}(t)$ at time instance t. The closest pattern update stops when the algorithm has converged with respect to a convergence threshold γ. That is, the patterns' updates are stopped at the first time instance (vector observation) t' such that:

$$t' = \inf_t \{t > 0 : \Gamma(t) \leq \gamma\}. \tag{33}$$

During the training phase, at every observation $\mathbf{x}(t)$ at time instance t, the algorithm finds the closest pattern $\mathbf{w}^*$ to the context vector $\mathbf{x}(t)$. This requires $O(dK)$ time per observation for searching for the closest pattern out of the current K patterns $\{\mathbf{w}_k\}_{k=1}^K$. The whole training process requires $O(dKt')$ time. After convergence, i.e., at time instance $t > t'$ the structures of the patterns are used for outliers detection and, in certain cases, for adaptation based on the optimal stopping time methodology in Section 5. In this phase, the calculation of the probability $p(\mathbf{w}^*|\mathbf{x})$ requires $O(d \log K)$ given a k-d tree structure for searching the closest pattern. The space complexity of Algorithm 1 refers to the storage of the K d-dimensional patterns $\mathbf{w}_k$, which is $O(dK)$.

9.1.2 Time & space complexity for change detection and adaptation

The Algorithm 2 is an incremental algorithm, which processes the sensed context vector $\mathbf{x}(t)$ at time instance t to determine whether there is a context change detection after several observations. Algorithm 2 requires a pre-calculation of the K scalars $\theta_k, k \in [K]$ using (22). Those scalars derive from the variances of the data subspaces represented by the patterns $\mathbf{w}_k$ from Algorithm 1. This requires $O(dK)$ time for the K variances θ_k. The Algorithm 2, at every time instance t, calculates the outlier indicator I_t using (16), which requires $O(1)$, given the closest pattern $\mathbf{w}^*$, which requires $O(d \log K)$. When the optimal stopping criterion holds true, which is determined in $O(1)$ time, then the closest pattern $\mathbf{w}^*$ is either updated or a new pattern is inserted in the pattern set $\mathcal{W}$ in $O(1)$. In the adaptation, the dynamic vigilance ρ^* is updated in $O(dK)$. The space complexity of Algorithm 2 refers to the storage of the K scalars (variances) θ_k and the K dynamic vigilances ρ_k^*, which is $O(K)$.

Table 2 shows the asymptotic time and space complexities for the Algorithms 1 and 2.

9.1.3 Time & space complexity for degree of belief inference

Each node i upon sensing a d-dimensional context vector $\mathbf{x}(t)$ at time instance t performs event inference to derive

Table 2 Asymptotic time & space complexities per node

Process	Time	Space
Vector Quantization (training)	$O(dKt')$	$O(dK)$
Vector Quantization (probability assignment)	$O(d \log K)$	
Change Detection (scalars θ_k)	$O(dK)$	$O(K)$
Change Detection (outlier indicator)	$O(1)$	
Context Adaptation	$O(1)$	$O(K)$
Context Prediction	$O(dm)$	$O(dm)$
Context Fusion	$O(d)$	$O(1)$
Fuzzy Inference	$O(R)$	$O(R)$

locally the degree of belief μ_i. Specifically, context prediction scales linearly with the number of the temporal nearest neighbors $h \leq m$ for smoothing, thus, requiring $O(dm)$ time to predict context. In addition, concerning the outliers detection, upon reception of a context vector, node i performs a nearest neighbor search over the K patterns to find the closest one. By adopting a d-dimensional tree structure (a k-d tree) over the prototypes, we require $O(d \log K)$ for evaluating the probability of assignment. In the case of adaptation after the outliers detection, node i adapts its closest pattern in $O(1)$. Moreover, the context fusion is achieved in $O(d)$ to evaluate the vector state, that is, it depends only on the data dimensionality. Finally, the FIRs are fixed and provided by the experts. Hence, the fuzzy-based event inference takes $O(R)$, where R is the number of FIRs.

Overall, based on Table 2, a node i requires $O(d(m + \log K) + R)$ to provide the degree of belief μ_i on a event including any possible adaptation. It is worth noting that, node i requires $O(d(K + m))$ space to store the patterns and the most recent context vectors. Given the belief-centric clustering, a node i after local inference can initiate a clustering era for determining the aggregated degree of belief. In each clustering era, every node i requires $O(1)$ messages to either be appointed as a CH or not (member of the cluster); see Lemmas 1 & 2. For a CH node, the calculation of the aggregated degree of belief depends on the cardinality of its neighborhood, i.e., number of cluster members, which requires $O(|\mathcal{N}|)$ using (30). Every CH node then transmits to its concentrator the aggregated degree of belief requiring $O(1)$ message (network communication). Table 3 summarizes the overall asymptotic complexities per node for the engaged processes: event inference, belief-centric election and report of the aggregated degree of belief to the concentrator.

9.2 Communication cost & computation energy consumption

The nodes must accomplish their assigned sensing and inference tasks by using the limited energy resources carried by them. The energy refers to a number of operations: (i) wireless communication, (ii) sensing the environment, and (iii) local computation. In our study, the energy and communication model reflects three facets: energy for communication required for the belief-centric clustering process, energy for computation, i.e., event inference and degree of belief derivation, and communication energy of the cluster-heads to report the aggregated degree of belief to their assigned concentrators.

Each node i consumes processing power for locally inferring the degree of belief μ_i of a possible event as described in Section 7.2. We notate with $\mathcal{E}_{\mu_i}$ the energy cost in Joule per CPU instructions corresponding to the executable inference algorithm for local degree of belief per node i. Moreover, when nodes initiate a clustering era, then some nodes are appointed as cluster-heads computing their EP values. During a clustering era, a node is either dynamically appointed as a CH or acting as a member. In a clustering era, the energy for in-cluster communication $\mathcal{E}_{c,i}$ in Joules per bit transmission (TX) and reception (RX) is the energy consumption incurred on node i by transmitting (TX) and receiving (RX) election messages. After the election, each CH node has to calculate its neighbors' aggregate belief $\tilde{\mu}_i$ with energy cost $\mathcal{E}_{\tilde{\mu}_i}$ in Joule per CPU instructions and then transmit (TX) this value to its assigned concentrator, thus, incurring an additional communication cost $\mathcal{E}_{CH,i}$.

Let the CH indicator $J_i = 1$ if node i is appointed as a CH after clustering era; otherwise $J_i = 0$ when node i is a cluster member. Then, we define the total cumulative energy consumption C_i per node i as the cumulative computation consumption for event inference and/or aggregated degree of belief, and communication consumption for clustering and transmitting the aggregated degree of belief to the concentrators (in the case of CHs only) up to time instance t, that is:

$$C_i = C_{p,i} + C_{c,i}, \tag{34}$$

where

$$C_{p,i} = \sum_{\tau=0}^{t} \left(\mathcal{E}_{\mu_i}^{\tau} + J_i \mathcal{E}_{\tilde{\mu}_i}^{\tau} + \mathcal{E}_0^{\tau} \right), \tag{35}$$

and

$$C_{c,i} = \sum_{\tau=0}^{t} \left(\mathcal{E}_{c,i}^{\tau} + J_i \mathcal{E}_{CH,i}^{\tau} + \mathcal{E}_0^{\tau} \right), \tag{36}$$

where $\mathcal{E}_0$ is the energy cost for node i transiting from idle to standby operational modes [36]. Up to time instance t, the communication and computation costs for all nodes and the overall cost are, respectively:

$$C_c = \sum_{i=1}^{|\mathcal{N}|} C_{c,i}, C_p = \sum_{i=1}^{|\mathcal{N}|} C_{p,i}, C = C_p + C_c. \tag{37}$$

Table 3 Asymptotic complexities for each process per node; '-' means 'not applicable'

Process	Node type	Computation	Communication
Election	member	$O(1)$ (election probability)	$O(1)$ (clustering era)
	cluster-head	$O(1)$ (election probability)	$O(1)$ (clustering era)
Event Inference	member	$O(d(m + \log K) + R)$	–
	cluster-head	$O(d(m + \log K) + R)$	–
Concentrator Report	member	–	–
	cluster-head	$O(\lvert\mathcal{N}\rvert)$ (aggregated $\tilde{\mu}$)	$O(1)$

For the sensing, communication and computation energy consumption, we adopted the energy model from the Mica2 sensor board.[4] This energy model assumes an energy of two AA batteries that approximately supply 2200 mAh with effective average voltage 3V. It consumes 20mA if running a sensing application continuously. The communication cost for transmitting (TX) a bit is 720 nJ/bit and receiving (RX) a bit is 110 nJ/bit. Moreover, the packet header of the communication protocol adopted by Mica2 is 9 bytes (MAC header and CRC) and the maximum payload is 29 bytes. Therefore, the per-packet overhead equals to 23.7% (lowest value). For each transmitted data value, i.e., a value component x of a d-dimensional vector $\mathbf{x}$ and the EP value in an election message, the assumed payload is set to 4 bytes (floating point number) and 2 bytes, respectively. Finally, the energy cost for single CPU instructions (energy per instruction) is 4 nJ/instruction in Mica2. Table 4 shows all the energy consumption in nJ per bit, for communication, and in nJ per CPU instruction, for computation.

10 Performance evaluation

10.1 Performance metrics

We assess the performance of our mechanism in terms of: (i) probability of false (erroneous) event inference $\phi \in [0, 1]$, (ii) event time index $\tau \in \mathbb{T}$ of recognizing an event, (iii) communication overhead (number of aggregated degree of belief messages) $\mathcal{M}$ required for CHs to inform the concentrators for event inference, (iv) energy consumption for event inference C_p and communication cost C_c per node i and the total IoT environment, and (v) efficiency of our mechanism in delivering event inference with a low false rate being communication and energy aware.

The false probability ϕ represents the rate of erroneous inference (false alerts) that the mechanism generates defined as the ratio of the number of false alerts out of a total number of inference results. Note, event inference is obtained at every time $t \in \mathbb{T}$ corresponding to the reception of context vector $\mathbf{x}$ at any node i. A value of $\phi \to 1$ indicates high rate of false alerts, thus, no conclusion can be drawn for the true state of the phenomenon.

The event time index $\tau \in \mathbb{T}$ refers to the time index of the measurement that actually corresponds to an event. Through that metric, we assess how 'close' to the real case an event is inferred by our mechanism; not at early stages in order to avoid false alerts and not many stages after the real event. The τ is evaluated by the rate of the identification for real events.

The number of messages $\mathcal{M}$ refers to the total number of messages ($\tilde{\mu}$ values) sent from CHs to their concentrators including the total number of messages sent for the belief-centric clustering. The lower the $\mathcal{M}$ is, the lower energy resources in terms of communication are spent. Let us notate the lifetime of the entire network as $\mathcal{T}$ (in terms of energy) and $\mathcal{N}_{CH}$ be the set of CHs, i.e., $|\mathcal{N}_{CH}| \ll |\mathcal{N}|$. Since at each clustering era $T_1, T_2, \ldots$, our mechanism assigns certain nodes as CHs then, in the network lifetime, $\lfloor \frac{\mathcal{T}}{T} \rfloor$ clustering eras are realized, where T is the expected number of clustering initiations out of the total number of observations. By adopting our belief-centric clustering, only $\mathcal{N}_{CH}$ messages of $\tilde{\mu}$ values are delivered to the concentrators to keep the concentrators up-to-date about the event inference along with $O(|\mathcal{N}|)$ messages circulated locally for building the clusters as proved in Lemma 2. Hence, it holds true that:

$$\mathcal{M} = \lfloor \frac{\mathcal{T}}{T} \rfloor \left(|\mathcal{N}_{CH}| + O(|\mathcal{N}|)\right).$$

Without clustering, all nodes would send their μs to the concentrators, thus, in this case we would obtain $\mathcal{M} = \lfloor \frac{\mathcal{T}}{T} \rfloor |\mathcal{N}|$.

The energy consumption C_p refers to the energy consumed for computational processing per node i to locally infer the degree of belief after observing a d-dimensional context vector. The energy consumption C_c refers to the

Table 4 Energy parameters

Parameter	Default values
Transmitting a bit energy consumption	720 nJ/bit
Receiving a bit energy consumption	110 nJ/bit
CPU energy consumption per instruction	4 nJ/instruction

[4] http://www.tinyos.net/scoop/special/hardware#mica2platform

communication overhead cost for nodes during the clustering eras due to messages exchange for CH election. These messages include the EP values. Moreover, this cost includes the energy consumption for the appointed CHs to transit the aggregated degrees of belief (from their neighborhood) to their concentrators. The energy model for computation and communication derives from the Mica2 energy model presented in Section 9.2. Finally, we define as *efficiency* the total amount of energy consumed from our mechanism $C = C_p + C_c$ to deliver event inference with a low false rate ϕ. We desire to obtain a low energy expenditure along with a low false rate. We compare our mechanism with other mechanisms in terms of energy consumption (communication and computation) and efficiency, as shown in Section 10.5.

10.2 Experiment setup

We experiment with a real multivariate dataset [38] adopted from the Microsoft research open datasets.[5] The dataset contains meteorological data retrieved in the cities of Beijing and Shanghai. The collected context variables are: temperature, humidity, barometer pressure and wind strength. In our experiments, we adopt 2-dim. context vectors with $x_1 =$ 'temperature' and $x_2 =$ 'humidity' recorded by $|\mathcal{N}| = 50$ nodes deployed in the field and observe 50,000 context vectors. We consider one observation at each discrete time instance $t \in \mathbb{T}$ and assume one concentrator acting also as the back-end system for those nodes. All vectors are scaled, i.e., $\mathbf{x} \in [0, 1] \times [0, 1]$.

In the dataset, no hazardous events are identified, i.e., the probability of a true event is zero. To define an event, we exploit the expert knowledge in [25] stating that: a high temperature, e.g., around 600 Celsius, along with a low humidity, e.g., below 30%, defines a fire incident. Firstly, we consider injecting 'faulty' values to examine whether our mechanism produces erroneous inference/false alerts. Our target is to obtain $\phi \to 0$. To simulate a setting where nodes deliver faults/outliers, we randomly inject faulty measurements as indicated by the 'faulty rules' in [26] with some fault probability $p_F > 0$. On a node i, an actual temperature value x_1 at time t will be replaced as $x_1 \leftarrow (1 + a_F)x_1$ and for humidity $x_2 \leftarrow \frac{a_F}{1+a_F}x_2$, with $a_F \in \{2, 3, 5\}$ and assume different faulty probabilities $p_F \in \{5\%, 10\%, 20\%, 40\%, 60\%, 80\%\}$. In addition, we inject a set of fire events represented by a state temperature value v_1 close to 1 and a state humidity value v_2 close to zero as depicted in [25]. Note, we increase the temperature value and decrease the humidity value corresponding to the same context vector. The event time index τ_k of a predefined fire event E_k is pre-recorded. We define 10 fire events randomly spanned in the dataset where: the time duration of an event is drawn from the Exponential distribution with average time event-duration 10 time units. Through this setup, we examine whether our mechanism is capable of (i) inferring the events E_k given fault probability p_F and (ii) producing a time index of E_k as close to τ_k as possible, i.e., if the proposed mechanism identifies E_k at the right time.

The parameter values are presented in Table 5 and, specifically the **default values** are: belief threshold $\epsilon = 0.7$, convergence threshold $\gamma = 0.001$, context history $m = 10$, $h = 5$ in hNN DES, vigilance percentage $a = 0.1$ and vigilance threshold is $\rho = (ad)^{1/2} = 0.44$ for 2-dim. context, initial learning rate $\eta = 0.5$, assignment probability factor $\beta = 0.1$, risk factor $\alpha = 0.95$, opinion factor $r = 0.5$, and the number of FIRs is $R = 27$. The justification of those values is discussed in the remainder.

10.3 Comparison models

We compare our mechanism, hereinafter referred to as Model (M), with the *local* Voting Scheme (VS) and the *centralized* Aggregation Scheme (AS).

In the local VS model, a node i locally infers an event at time t based only on the expert knowledge fusion function, i.e., when it holds true that:

$$f(\mathbf{v}_i(t)) \geq \epsilon, \tag{38}$$

thus, neglecting all other CPs to reason about the final decision. Then, each node i transmits *only* its inference result (event vote) to a central node gathers, which centrally infers an event based on the majority of votes.

In the centralized AS model, each node i transmits its current context data vector $\mathbf{x}_i(t)$ to the central node. The central node, then, aggregates all the received context data

Table 5 Experimental parameters

Parameter	Values
Number of nodes $\|\mathcal{N}\|$	{5, 10, 50}
Number of *actual* events	10
Number of observations per node T	10,000
Data Faulty Probability p_F	{5%,10%, 20%, 40%, 60%, 80%}
Belief threshold ϵ	{0.5, 0.7, 0.9}
Convergence threshold γ	0.001
Context history for prediction (m,h)	(10,5) hNN DES
Vigilance percentage a	{0.05, 0.1, 0.3, 0.5, 0.7, 0.9}
Assignment probability factor β	0.1
Risk factor α	0.95
Opinion factor r	0.5
Number of FIRs R	27

[5] http://research.microsoft.com/en-us/projects/urbancomputing/default.aspx#datasets

Table 6 Model M: false rate ϕ vs. p_F and $|\mathcal{N}|$

	ϕ		
p_F	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$
5%	0.001	0.000	0.000
10%	0.005	0.000	0.000
20%	0.011	0.002	0.000
40%	0.013	0.004	0.000
60%	0.014	0.004	0.000
80%	**0.015**	**0.005**	**0.003**

vectors from the $|\mathcal{N}|$ nodes and *centrally* infers an event based on:

$$f(g\{\mathbf{v}_i(t)\}_{i=1}^{|\mathcal{N}|}) \geq \epsilon, \tag{39}$$

where $\mathbf{v}_i(t)$ is the state context vector corresponding to node i's context and $g\{\cdot\}$ is the average operator.

10.4 Performance evaluation

10.4.1 Quality of event inference

We analyze the event inference performance of model M for different values of nodes $|\mathcal{N}|$, faulty probabilities p_F, fusion parameter λ_2, belief threshold ϵ, and vigilance ρ. In Table 6, we examine the *robustness* of model M in terms of false rate ϕ for different values of faulty probability p_F, $5\% \leq p_F \leq 80\%$, and number of nodes $|\mathcal{N}|$. Model M is robust assuming a very low ϕ (less that 1.5%) even for data streams involving a huge number of faulty values i.e., $p_F = 80\%$. This indicates the capability of model M to reason under uncertainty as treated by the involvement of the three CPs. Moreover, the knowledge fusion of the local degrees of belief depends on the number of opinions, i.e., the number of nodes involved in the event reasoning. The higher the $|\mathcal{N}|$ is, the lower the ϕ becomes. The reason is that each node i locally process context and infers an even w.r.t. the three CPs and shares its local view/degree of belief with its neighbors through our CH-based consensus approach. Then, by voting among those aggregated degrees of belief $\tilde{\mu}$, which actually related to an event (sent only by CHs), the back-end system clearly concludes on that event with high accuracy. Model M takes into consideration the groups' perspectives, i.e., an event is locally agreed on a CH only when a large percentage of neighboring nodes support that event presence. When $|\mathcal{N}|$ increases, the team is more 'compact' meaning that much more nodes support an event presence with more certainty in contrast to the case where $|\mathcal{N}|$ is small. In the case where only one node i is present, model M is based on node i's belief, thus, false alerts could arise more easily (as node i could have a faulty view on an event presence).

In addition, we examine the impact of $|\mathcal{N}|$ on the time lag τ from the actual event time index and the identified/inferred time index. Model M obtains an average time lag $\tau = 2.2$ time units with standard deviation $\sigma_\tau = 0.77$ for $5 \leq |\mathcal{N}| \leq 50$. This indicates that all events are identified in very near real-time.

Table 7 presents the effect of the expert knowledge fusion (CP1) on producing false alerts. Recall that expert knowledge fusion depends on parameters λ_1 and λ_2 that affect the result for $f(\mathbf{v})$. From these two parameters, λ_1 'defines' the threshold value of the fusion function as provided by the expert, while λ_2 defines the steepness of the function. We experiment with the steepness $\lambda_2 \in \{2.0, 4.0, 6.0\}$ for fixed threshold λ_1. We observe in Table 7 that a high λ_2 results to a high false rate ϕ, while when $\lambda_2 = 2.0$, false rate ϕ is limited (equal or very close to 0). A high λ_2 leads to a more 'relaxed' identification of the event. However, this leads to an increased number of false alerts by overestimating the CP1 $f(\mathbf{v})$ at the expense of the other two CPs (error e and assignment probability $p(\mathbf{w}^*|\mathbf{x})$), which is passed to the T2FLS engine. A low λ_2 value regulates the impact of CP1 on the other two CPs, thus, model M exploits all CPs to avoid high rate of erroneous inference results.

We also examine the impact of the belief threshold ϵ on model M in terms of false rate ϕ. Table 8 shows the results

Table 7 Model M: false rate ϕ vs. p_F and $|\mathcal{N}|$ for $\lambda_2 \in \{2.0, 4.0, 6.0\}$

	ϕ with $\lambda_2 = 2.0$			ϕ with $\lambda_2 = 4.0$			ϕ with $\lambda_2 = 6.0$		
p_F	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$
5%	0.000	0.000	0.000	0.157	0.046	0.001	0.370	0.252	0.010
10%	0.000	0.000	0.000	0.164	0.075	0.001	0.476	0.359	0.031
20%	0.000	0.000	0.000	0.175	0.116	0.005	0.488	0.472	0.172
40%	0.001	0.000	0.000	0.182	0.210	0.072	0.496	0.483	0.548
60%	0.003	0.001	0.001	0.208	0.217	0.080	0.570	0.650	0.609
80%	0.003	0.002	0.001	0.301	0.357	0164	0.662	0.801	0.817

Table 8 Model M: false rate ϕ vs. p_F and $|\mathcal{N}|$ for $\epsilon \in \{0.5, 0.9\}$

	ϕ with $\epsilon = 0.5$			ϕ with $\epsilon = 0.9$		
p_F	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$	$\|\mathcal{N}\| = 5$	$\|\mathcal{N}\| = 10$	$\|\mathcal{N}\| = 50$
5%	0.881	0.819	0.773	0.001	0.000	0.000
10%	0.887	0.921	0.933	0.003	0.000	0.000
20%	0.947	0.971	0.972	0.005	0.001	0.000
40%	0.942	0.982	0.986	0.010	0.004	0.000
60%	0.945	0.980	0.982	0.011	0.008	0.001
80%	0.926	0.993	0.993	0.012	0.008	0.001

when $\epsilon \in \{0.5, 0.9\}$ for different values of $|\mathcal{N}|$ and p_F. A low ϵ leads to an optimistic and sensitive event identifies compared to high ϵ values. Evidently, this corresponds to an increased number of false alerts ϕ. In such cases, model M is also affected by an increased number of messages $\mathcal{M}$ sent from CHs to the back-end system, which reaches the theoretical maximum $\mathcal{M}$: the centralized approach, where all nodes send their observations to a back-end system. In addition, in Table 8 we observe results for $\epsilon = 0.9$. In this case, ϕ is minimized especially when $|\mathcal{N}| > 5$. A high ϵ makes event inference more insensitive and difficult to discriminate, thus, a limited number of nodes agree on an event presence. This behavior has obvious consequences on the identification of real events as τ is getting high. We set $\epsilon = 0.7$ in our experiments as explained later.

In addition, we experiment with the average number of context patterns K per node that are required to quantize the vector data space to materialize the CP2 and CP3. Table 9 shows the number of K patterns (mean value avg(K) and standard deviation σ_K out of $|\mathcal{N}| = 50$ nodes) that quantize the context spaces needed for outliers and novelty detection against the vigilance percentage a, i.e., $\rho = (ad)^{1/2}$. A low a value, which corresponds to low ρ, results in high quantization resolution in terms of patterns; a high number of patterns are generated to better represent the vector space. This, however, comes at the expense of a high number of patterns that are needed to be stored on a node. But, even in the case of $a = 0.1$, this number is significantly low ($K \sim 49$). Hence, to achieve highly accurate inference results and maintain the model M up-to-date w.r.t. novelty vector subspaces, we set $a = 0.1$.

10.4.2 Communication & computation cost

In terms of communication overhead (number of messages circulated in the IoT environment), we examine the capability of model M to achieve low false rates by avoiding transferring context data to the concentrator, but only the minimal sufficient knowledge for event reasoning in terms of belief threshold ϵ. Table 9 shows the impact of belief threshold ϵ on: (i) the number of messages $\mathcal{M}$, (ii) the average number of CHs $|\mathcal{N}_{CH}|$ per clustering era, and (iii) the number of clustering eras T. A value of ϵ close to the cut-off value of 0.5 results to many clustering eras ($T > 1000$), thus, many messages are sent between clusters and from CHs to concentrators along with high ϕ value (see Table 8). Evidently, a value $\epsilon > 0.5$ is adopted to 'narrowing' and clarifying the inference results. On the other hand, with a high ϵ, model M increases its tolerance to assess an event presence thus being communication efficient. However, in this case, events are difficult to identify, which does not reflect the actual situation on the IoT network. To balance between communication load, accuracy of inference, and capability of event identification, we set a belief threshold $\epsilon = 0.7$ in our experiments. For $\epsilon = 0.7$, model M initiates $T = 10$ clustering eras in which 9% of nodes (CHs) transfer their aggregated knowledge to concentrators achieving a low ϕ value. In all these 10 clustering eras, the nodes successfully detect all 10 events.

In terms of energy consumption due to the computational cost of local inference and communication cost for each clustering era, we present in Fig. 2 (left) the total cost C_M in nJoule and its breakdown in the processing

Table 9 Model M: patterns K per node vs. $\rho(a)$; Messages $\mathcal{M}$, average number of cluster heads $|\mathcal{N}_{CH}|$, and clustering eras T vs. belief threshold ϵ; $|\mathcal{N}| = 50$

a	$\rho(a)$	avg(K)	σ_K	ϵ	$\mathcal{M}$	avg($\|\mathcal{N}_{CH}\|$)	T
0.05	0.31	72.18	8.76	0.50	$1.20 \cdot 10^4$	19.20	1247
0.10	**0.44**	**49.12**	**5.64**	0.60	$8.11 \cdot 10^3$	11.67	284
0.30	0.77	13.56	3.21	0.70	$\mathbf{3.42} \cdot 10^3$	**4.65**	**10**
0.50	1.00	08.77	2.42	0.75	$1.76 \cdot 10^3$	3.87	8
0.70	1.18	05.54	1.02	0.80	$5.12 \cdot 10^2$	2.33	3
0.90	1.34	01.47	0.74	0.90	$2.49 \cdot 10^2$	1.46	2

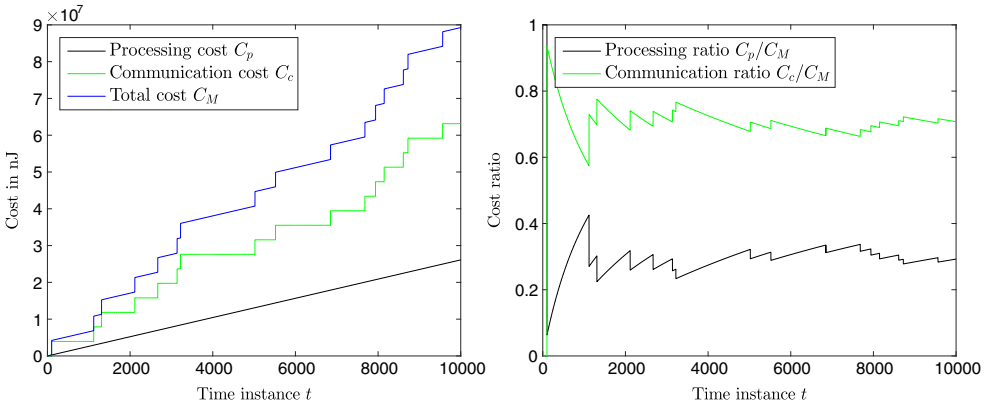

Fig. 2 (*Left*) Total energy consumption C_M in nJ and its breakdown to the processing/computation cost C_p and communication cost C_c for $|\mathcal{N}| = 50$ nodes vs. number of observations; (*right*) the processing and communication ratios out of the total consumed energy

(computation) cost C_p and communication cost C_c, as defined in (34) and (37), respectively, for $|\mathcal{N} = 50|$ nodes, after observing 10,000 context vectors each. We can observe that the computational consumed energy is the lowest energy expenditure compared to the communication cost, which indicates the advantage of the distributed inference, thus reducing the network overhead. This is attributed to the fact that the energy required for TX and RX pieces of data is higher than the energy consumed for local computations in each node. Moreover, our proposed Algorithms 1 and 2 are on-line, incremental learning algorithms, thus providing a lightweight solution for localized context inference, which is our major goal: 'to push on the intelligence to the edge of the network', reducing unnecessary data transfer to the back-end system and/or to the concentrators. Since each node i can locally reason about contextual event then, instead of transmitting *actual* sensed contextual multidimensional data towards a centralized system (as it will be discussed later in the comparative assessment Section 10.5), it transmits *only* if needed the local inference results, i..e, the degree of belief. Moreover, the proposed clustering scheme involves localized message exchange among neighboring nodes, which further reduces the network overhead by avoiding transiting data values from the edge of the network to the concentrators. Even in this localized information dissemination process, the nodes are transmitting only inferred knowledge, i.e., local degrees of belief and not data values. The appointed CHs are the only responsible ones to transmit the aggregated degrees of belief to the concentrators, where they corresponds to the 9% of the total number of nodes in the network. By pushing this intelligence to the edge, the computational cost consists of 30% of the total consumed energy, while the remaining portion is devoted to localized communication during the clustering eras plus the communication of the CHs with the concentrators, as shown in Fig. 2 (right).

Figure 3 (left) shows the total consumed energy C_M for different number of nodes $|\mathcal{N}|$, while Fig. 3 (right) illustrates the impact of the vector quantization (vigilance percentage a) in each node i on the processing/computation cost C_p out of the total cost C_M for different number of nodes. It is worth mentioning that when we increase the resolution of the vector quantization, i.e., the number of patterns K that can be estimated during the vector quantization process (Algorithm 1) then the node i spends more energy for computation. This corresponds to identifying the closest pattern and to calculate the assignment probability. Obviously, the more patterns each node derives from

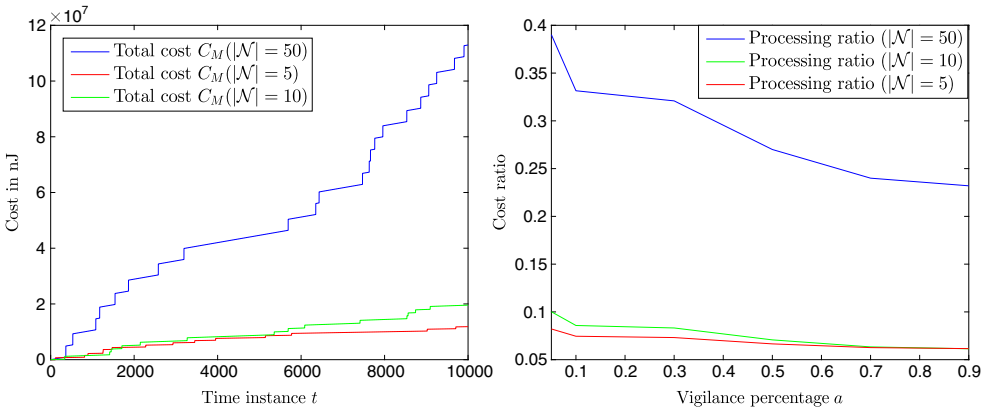

Fig. 3 (*Left*) Total energy consumption C_M in nJ for different number of nodes $|\mathcal{N}|$ nodes vs. number of observations; (*right*) The processing/computation cost ratio C_p/C_M vs. the quality of vector quantization (vigilance percentage a) for different number of nodes $|\mathcal{N}|$

the quantization process the higher the quality of inference, however, at the expense on the computational energy consumption. Nonetheless, the quality of inference is related with reducing the false rate ϕ. By achieving a significant low ϕ value, i.e., $\phi < 0.001$, our mechanism requires a vigilance percentage $a = 0.35$. In this case, the processing ratio is approximately 30% of the total energy consumption. There is then a trade-off between quality of inference (due to high quality of vector quantization) and required energy for achieving this high quality. Our mechanism is flexible to tune this trade-off (as shown in Fig. 6) and attempts the lowest false rate by being energy efficient (in both: communication and computation) compared with the VS and AS models described in Section 10.5.

10.5 Comparative assessment

We compare model M with the models VS and AS, where their inference policies are provided in (38) and (39), respectively, focusing on: (i) quality of event inference, (ii) energy consumption in terms of computational cost and communication overhead, and (iii) efficiency.

10.5.1 Comparison in quality of inference

In the quality of inference we evaluate the false rate for each model given a probability of faulty data values to examine their robustness. In the comparison experiments, we take $|\mathcal{N}| \in \{5, 10, 50\}$. Table 10 shows the false rate ϕ for $|\mathcal{N}| \in \{5, 10, 50\}$ and different p_F values. We observe that model M outperforms VS and AS models when $p_F \geq 40\%$. This is interesting as it shows that model M achieves a bounded erroneous inference probability even when nodes experience multiple faulty measurements. For $p_F = 80\%$ indicating high uncertainty, model M achieves 80.00% and 82.76% fewer false alerts compared to VS and AS, respectively. We can also observe from Table 10 the comparison results for $|\mathcal{N}| \in \{10, 50\}$. In general, the increased number of nodes leads to a low number of false alerts (i.e., low ϕ), close to zero. For $|\mathcal{N}| = 10$, model M outperforms VS and AS when $p_F > 40\%$. For $p_F = 80\%$, model M achieves 88.10% and 84.85% fewer false alerts compared to VS and AS, respectively. In any case, model M keeps ϕ close to zero. This indicates the capability of model M to exploit all CPs to reason about event in a robust way along with taking into account the local degrees of belief of neighboring nodes. For $|\mathcal{N}| = 50$, model M produces alerts with very high accuracy, i.e., low ϕ, compared with models AS and VS, for all p_F values.

10.5.2 Comparison in energy consumption, cost & efficiency

Model M obtains significant low false rates with a significant low number of messages sent from CHs to the back-end systems, compared to the VS and AS models. Specifically, for model M there are $T = 13$ clustering eras out of the total $5 \cdot 10^4$ observations, and we obtain number of messages $\mathcal{M} = (2.32 \cdot 10^6, 24.6 \cdot 10^4, 3.42 \cdot 10^3)$ for model AS, model VS and model M, respectively (we obtain, on average, $|\mathcal{N}_{CH}| = 4.65$ cluster heads per clustering era). This indicates that, for even uncertain and faulty data streams, i.e., $p_F > 40\%$, model M achieves 83.67% lower false rate from both AS and VS models by requiring three and one less orders of magnitude in communication overhead, respectively.

In terms of energy consumption in computation and communication, Fig. 4 (left) show the total cost C_M, C_{VS}, and C_{AS} for model M, VS, and AS, respectively, for $|\mathcal{N}| = 50$ in logarithmic scale. It is obvious that our model saves energy by at least two orders of magnitude compared to the localized inference model VS and the centralized inference model AS. This indicates the vision of pushing intelligence to the edge of the network with exploiting the computing capability of the nodes to infer events, thus avoiding data transfers from the source of information to the back-end-systems. Moreover, even in the case of the localized VS model, our model requires significantly less energy (two orders of magnitude) since the 'instant' inference achieved by a node executing the VS model appears to be in many

Table 10 Comparison: model M, VS, and AS for ϕ vs. p_F, $|\mathcal{N}| \in \{5, 10, 50\}$

	$\|\mathcal{N}\| = 5$			$\|\mathcal{N} = 10\|$			$\|\mathcal{N}\| = 50$		
p_F	ϕ_M	ϕ_{VS}	ϕ_{AS}	ϕ_M	ϕ_{VS}	ϕ_{AS}	ϕ_M	ϕ_{VS}	ϕ_{AS}
5%	0.001	0.001	0.000	0.000	0.007	0.005	0.000	0.007	0.004
10%	0.005	0.000	0.000	0.000	0.011	0.015	0.000	0.012	0.010
20%	0.011	0.002	0.002	0.002	0.018	0.016	0.000	0.019	0.015
40%	0.011	0.012	0.011	0.004	0.028	0.018	0.000	0.023	0.019
60%	0.014	0.042	0.038	0.004	0.038	0.029	0.000	0.035	0.026
80%	**0.015**	0.075	0.087	**0.005**	0.044	0.039	**0.003**	0.055	0.029

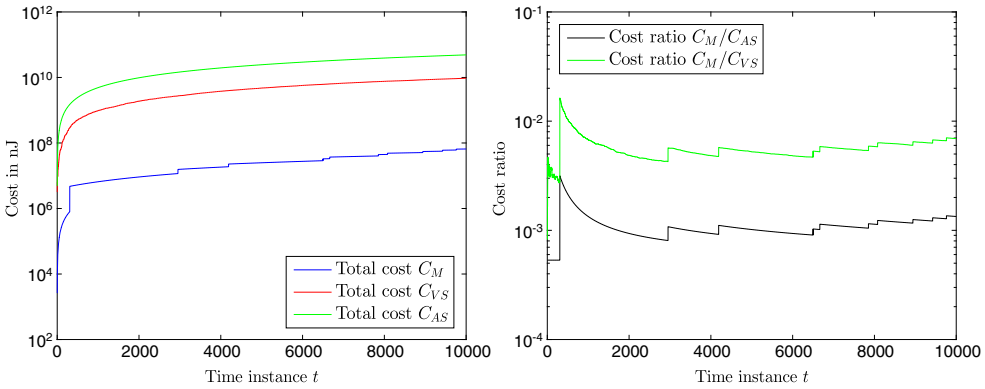

Fig. 4 (*Left*) Total energy consumption for models M, VS, and AS vs. number of observations; (*right*) the cost ratios C_M/C_{AS} and C_M/C_{VS} of the model M out of the models AS and VS, respectively vs. number of observations; $|\mathcal{N}| = 50$

times erroneous compared to our model. We capture this by introducing intelligent context reasoning processes such that the CHs will only infer an event if the neighboring nodes reach a consensus, thus minimizing the false rate. This, however, requires some additional computational cost and communication. But, as illustrated in Fig. 4 (right), the ratio of the consumed energy by our model is $\sim 10^{-3}$ and $\sim 10^{-2}$ of the consumed energy by the centralized and localized models, respectively.

In Fig. 5 (left) we examine the scalability capability of our model in terms of the number of CHs as a percentage of the total number of nodes comparing with the AS and VS models. Specifically, we present the total consumed energy (computation and communication) starting from a CH percentage $|\mathcal{N}_{CH}|$ of 10% to 100% of the total number of nodes $|\mathcal{N}|$. We can observe the significantly low impact on the total cost compared with the other models. Moreover, the case where $|\mathcal{N}_{CH}| = |\mathcal{N}|$ depicts the capability of inferring an event as accurately as possible by each of the nodes, thus minimizing the ϕ value. It is worth comparing this scalability performance with the VS model, where all the nodes are acting independently based on the inference policy in (38). This indicates the capability of our model not only to scale with the number of CHs but also to deliver inference results corresponding to high quality of inference.

Figure 5 (right) shows the impact of the belief threshold ϵ on the consumed energy for all the models with $|\mathcal{N}| = 50$. The higher the ϵ value the less insensitive each model is to accurately inferring an event. However, this comes at a lower cost, since both the model M and model VS avoid inferring events, thus, reducing the communication with the back-end-system (transmitting the inference results from the CH nodes in model M and from the individual nodes in model VS). Evidently, the model AS is not influenced by this threshold since the nodes just deliver the sensed contextual vectors and do not perform any computations. On the other hand, this results to a high impact on the quality of inference, which is quantified by the ϕ value. Given a

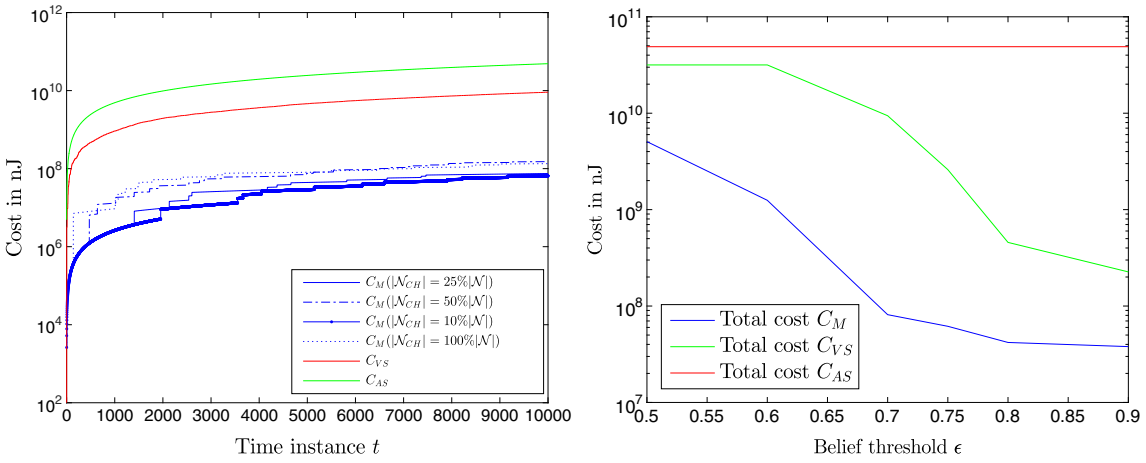

Fig. 5 (*Left*) Scalability: total energy consumption for models M, VS, and AS vs. number of observations. For model M the cost C_M is shown for different percentages of the number of CH nodes $|\mathcal{N}_{CH}|$ out of the total number of nodes $|\mathcal{N}| = 50$; (*right*) total energy consumption for models M, VS, and AS vs. the belief threshold ϵ with $|\mathcal{N}| = 50$

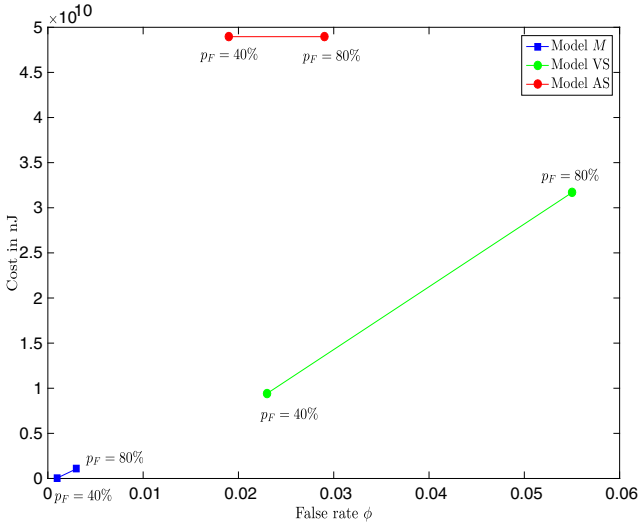

Fig. 6 Efficiency: total energy consumption for models M, VS, and AS vs. false rate ϕ for different faulty value probabilities p_F with number of nodes $|\mathcal{N}| = 50$

significant low value of ϕ, we set $\epsilon = 0.7$, which results to three orders and two orders of magnitude less energy consumption achieved by our model, comparing with the AS and VS models, respectively. In that case, we define the efficiency indicator to examine the consumed energy of each model and its corresponding performance in terms of quality of inference.

Figure 6 shows the total energy consumption for all models against false rate ϕ for data faulty probability $p_F \in \{40\%, 80\%\}$. It is worth noting the efficiency of our model compared to the other models AS and VS, which achieves very low false rate with significantly the lowest total energy consumption. The AS model appears to be the least efficient in terms of energy consumption and the achieved false rate, while model VS is moderate efficient for $p_F = 40\%$. When the faulty probability is high, then the model VS increases significantly its false rate, due to the lack of any reasoning algorithm to deal with high faulty data values, while it also consumes significantly more energy than model M. The model AS cannot reduce its energy consumption even if the data faulty probability decreases since that model does not take into consideration any characteristic of the captured contextual data streams (it only forwards data vectors to the back-end-system). Our model appears very robust in terms of efficiency even if the p_F is high. Overall, our concept of pushing predictive intelligence and data processing to the edge devices benefit: (i) accurate event inference close to the source of the information, (ii) significantly low communication overhead by localized belief-centric groupings, thus, avoiding data transfer to the back-end systems, and (iii) energy-efficient and robust inference in terms of data faulty probability.

11 Conclusions

We propose a novel federated event reasoning scheme by pushing predictive intelligence to the edge of the IoT network. This is achieved by an energy-efficient, real-time event reasoning mechanism, where data processing and predictive intelligence is pushed to the edge devices equipped with sensing and computing capabilities. Edge predictive intelligence and collaborative reasoning is materialized by the autonomous nature of nodes to locally perform data sensing & inference, and convey only inferred knowledge to their neighbors and concentrators. Nodes possess intelligence to reason about events, thus avoiding transferring raw data, while the complexity of inference is physically distributed to the sources of contextual information. Nodes are capable of locally processing and inferring events from contextual data streams enhanced with different context perspectives: predicted context, outliers context inference, and context fusion. The approximate event inference of each node is derived through Type-2 Fuzzy Logic inference to handle uncertainty. Finally, a knowledge-centric clustering scheme is introduced, where the clusters of nodes are formed according to their degrees of belief. The cluster heads are then disseminate the minimal sufficient knowledge to the concentrators / systems for event inference.

We provide mathematical analyses of our the statistical learning and stochastic optimization models, asymptotic complexities and energy consumption models for computation and communication cost, evaluate the model's performance and provide a comprehensive comparative assessment with other local & centralized event inference mechanisms. It is evidenced that the idea of exploiting the computing and sensing capabilities of nodes to 'intelligence at the edge' is deemed appropriate for real-time applications in IoT environments.

Appendix

A.1 Proof of Theorem 2

$\mathcal{J}$ is minimized by updating $\mathbf{w}_k$ in the negative direction of the sum of gradients, thus, obtaining $\Delta\mathbf{w}_k = -\eta\frac{\partial\mathcal{J}}{\partial\mathbf{w}_k} = \eta(\mathbf{x} - \mathbf{w}_k)$ by conditionally updating the closest $\mathbf{w}_k$. The $\mathbf{w}_k$ converges when $\mathbb{E}[\Delta\mathbf{w}_k] = \mathbf{0}$ given that $\|\mathbf{x} - \mathbf{w}_k\| \leq \rho$. We require at the convergence that $\mathbf{x}$ is assigned to its

closest $\mathbf{w}_k$ with probability 1, that is, $P(\|\mathbf{x}-\mathbf{w}_k\| \leq \rho) = 1$, which means that no other patterns are generated. Therefore, $P(\|\mathbf{x}-\mathbf{w}_k\| \geq \rho) \leq \frac{\mathbb{E}[\|\mathbf{x}-\mathbf{w}_k\|]}{\rho}$ or:

$$P(\|\mathbf{x}-\mathbf{w}_k\| \leq \rho) \geq 1 - \frac{\mathbb{E}[\|\mathbf{x}-\mathbf{w}_k\|]}{\rho}$$

based on Markov's inequality. To obtain $P(\|\mathbf{x}-\mathbf{w}_k\| \leq \rho) \to 1$ we have either $\rho \to \infty$ or $\mathbb{E}[\|\mathbf{x}-\mathbf{w}_j\|] \to 0$. However, ρ is a real small number, since it interprets the concept of neighborhood, then we require that $\mathbb{E}[\|\mathbf{x}-\mathbf{w}_j\|] \to 0$, i.e., $\mathbb{E}[(\mathbf{x}-\mathbf{w}_k)] = \mathbf{0}$ or $\mathbb{E}[\Delta\mathbf{w}_k] = \mathbf{0}$, which completes the proof.

A.2 Proof of Theorem 3

Let the pattern $\mathbf{w}_k$ reach equilibrium, i.e., $\Delta\mathbf{w}_k = \mathbf{0}$, which, in this case this holds with probability 1. Then, from the update rule in Theorem 2 by taking the expectation of both sides of $\Delta\mathbf{w}_j = \mathbf{0}$ at equilibrium we have that:

$$\mathbb{E}[\Delta\mathbf{w}_k] = \int_{\mathbb{D}_k} (\mathbf{x}-\mathbf{w}_k)p(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\mathbb{D}_k} \mathbf{x}p(\mathbf{x})\mathrm{d}\mathbf{x} - \mathbf{w}_k \int_{\mathbb{D}_k} p(\mathbf{x})\mathrm{d}\mathbf{x}.$$

By solving $\mathbb{E}[\Delta\mathbf{w}_k] = 0$, $\mathbf{w}_k = \bar{\mathbf{x}} = \int_{\mathbb{D}_k} \mathbf{x}p(\mathbf{x})\mathrm{d}\mathbf{x}$ i.e., the centroid of all vectors in $\mathbb{D}_k$.

A.3 Proof of Theorem 4

We have to prove that the optimal stopping time t^* exists and is derived from the principle of optimality: i.e., prove that (i) $\lim_{t\to\infty} \sup_t Y_t \leq Y_\infty$ a.s. and (ii) $\mathbb{E}[\sup_t Y_t] < \infty$. Note that I_t are non-negative and from the strong law of numbers $(\frac{1}{t})\sum_{\tau=1}^{t} I_\tau \to \mathbb{E}[I] = P(\{I=1\})$, so that:

$$Y_t = t\alpha^t(S_t/t) \leq t\alpha^t(1/t)\sum_{\tau=1}^{t} I_k \simeq t\alpha^t\mathbb{E}[I] \to 0,$$

with $\lim_{t\to\infty} \sup_t Y_t = Y_\infty = 0$. In addition,

$$\sup_t Y_t = \sup_t \alpha^t S_t \leq \sup_t \alpha^t \sum_{\tau=1}^{t} I_\tau \leq \sup_t \sum_{\tau=1}^{t} \alpha^\tau I_\tau \leq \sum_{\tau=1}^{\infty} \alpha^\tau I_\tau.$$

Hence, $\mathbb{E}[\sup_t Y_t] \leq \sum_{\tau=1}^{\infty} \alpha^\tau \mathbb{E}[I] = \mathbb{E}[I]\frac{\alpha}{1-\alpha} < \infty$. This completes the proof.

A.4 Proof of Theorem 5

Consider the indicators $I_1, I_2, \ldots$ with finite expectation $\mathbb{E}[I]$ and the random variables $Z_1, Z_2, \ldots$ such that $P(Z_t = 1) = \alpha$ and $P(Z_t = 0) = 1-\alpha$, $\alpha \in (0,1)$. We can then express our confidence as: $Y_t = \prod_{\tau=1}^{t} Z_\tau \cdot \sum_{\tau=1}^{t} I_\tau$, with $Y_\infty = 0$. Taking the expectation given $\mathbb{A}_t = \prod_{\tau=1}^{t} Z_\tau = 1$, i.e.:

$$\mathbb{E}[Y_{t+1}|\mathbb{A}_t] = \mathbb{E}\left[Z_{t+1}\sum_{\tau=1}^{t+1} I_\tau\right] = \alpha\left(\sum_{\tau=1}^{t} I_\tau + \mathbb{E}[I]\right),$$

then, the one-stage look-ahead rule from Theorem 4 is:

$$t^* = \min\left\{t \geq 1 : \sum_{\tau=1}^{t} I_\tau \geq \alpha\left(\sum_{\tau=1}^{t} I_\tau + \mathbb{E}[I]\right)\right\}$$

or $t^* = \min\{t \geq 1 : \sum_{\tau=1}^{t} I_\tau \geq \frac{\alpha}{1-\alpha}\mathbb{E}[I])\}$. This completes the proof.

A.5 Proof of Lemma 1

Consider a multiplication factor $\chi > 1$ and that node i starts with the minimum EP of being a cluster head, i.e., $\xi_i = \xi_{\min} > 0$. Since at each iteration step the node just multiplies its current EP ξ_i with χ then, in the worst case, that node will be either a cluster head or a member when the process stops at the first iteration step L such that $\chi^{L-1}\xi_{\min} \geq 1$. That is, the maximum number of iteration steps are $L = \min\{\ell > 0 : \chi^{\ell-1}\xi_{\min} \geq 1\}$. Hence, the required number of iterations is $L = \lceil \log_\chi \frac{1}{\xi_{\min}} \rceil + 1$, which maps to $O(1)$ iterations. Now, if node i starts the election process with $\xi_i > \xi_{\min}$ then $O(1)$ iterations are the maximum number of steps for the election process.

A.6 Proof of Lemma 2

In the election process, a node which is about to become a cluster head generates at most $L = O(1)$ messages. On the other hand, a node which is about to become a member delays in sending messages and sends one message to just join its cluster head after considering itself as 'non-cluster head'. Obviously, the number of those messages (member messages) is strictly less than $|\mathcal{N}|$, since at least one node will decide to be a cluster head. Hence, the number of messages exchanged in the network is upper-bound by $L \cdot |\mathcal{N}|$, which is $O(|\mathcal{N}|)$.

A.7 Analytical expression of $P(\{I=1\})$ using the Marcum Q-function

We have that: $P(\{I=1\}) = Q_{\frac{d}{2}}\left(\sqrt{\zeta}, \sqrt{\theta}\right)$. According to [40], the $Q_{\kappa_1}(\kappa_2, \kappa_3)$ is expressed in infinite series w.r.t the lower incomplete Γ function by substituting the $(\kappa_1, \kappa_2, \kappa_3) = (\frac{d}{2}, \sqrt{\zeta}, \sqrt{\theta})$ in our case, we obtain:

$$Q_{\frac{d}{2}}(\sqrt{\zeta}, \sqrt{\theta}) = e^{-\frac{\zeta}{2}} \sum_{k=0}^{\infty} \frac{\zeta^k}{2^k k!} \frac{\Gamma(k+\frac{d}{2}, \frac{\theta}{2})}{\Gamma(k+\frac{d}{2})}$$

where the lower incomplete function $\Gamma(z,x) = \int_x^\infty e^{-t}t^{z-1}dt$ is defined in [41]; equation (6.5.3) and Euler function $\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}dt$. Note, the discussed generalized Marcum Q-function is approximated using the Matlab function `marcumq(`$\kappa_2, \kappa_3, \kappa_1$`)` using the algorithm developed in [42].

References

1. Bourgeois W, Romain AC, Nicolas J, Stuetz RM (2003) The use of sensor arrays for environmental monitoring: interests and limitations. J Environ Monit 5(6)
2. Chehri A, Fortier P, Tardif PM (2007) Security monitoring using wireless sensor networks, 5th an. conf on communication networks and services research
3. Di Palma D, Bencini L, Collodi G, Manes A (2010) Distributed monitoring systems for agriculture based on wireless sensor network technology. Int J Adv Networks Services 3(1–2)
4. Hagras H, Doctor F, Callaghan V, Lopez A (2007) An incremental adaptive life long learning approach for type-2 fuzzy embedded agents in ambient intelligent environments. IEEE TFS 15
5. Dressler F, Nebel R, Awad A (2007) Distributed passive monitoring in sensor networks. INFOCOM
6. Fernández-Berni J, Carmona-Galán R, Martínez-Carmona JF, Rodríguez-Vázquez Á (2012) Early forest fire detection by vision-enabled wireless sensor networks. Int J Wildland Fire 21
7. Fisne A, Kuzu C, Hudaverdi T (2011) Prediction of environmental impacts of quarry blasting operation using fuzzy logic. Environ Monit Assess 174
8. Gouveia C, Fonseca A (2008) New approaches to environmental monitoring: the use of ICT to explore volunteered geographic information. GoeJournal 72
9. Hagras H (2004) A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE TFS 12
10. Hardas BM, Asutkar GM, Kulat KD (2008) Environmental monitoring using wireless sensors: a simulation approach, 1st int conference on emerging trends in engineering and technology
11. Hsin CF, Liu M (2002) A distributed monitoring mechanism for wireless sensor networks. WiSe
12. Kausar F, Al Eisa E, Bakhsh I (2012) Intelligent home monitoring using RSSI in wireless sensor networks. International Journal of Computer Networks and Communications 14(6)
13. Kolomvatsos K, Anagnostopoulos C, Hadjiefthymiades S (2015) An efficient environmental monitoring system adopting data fusion, prediction and fuzzy logic, 6th international conference on information, intelligence, systems and applications, Corfu, Greece
14. Kolomvatsos K, Anagnostopoulos C, Hadjiefthymiades S (2015) Intelligent contextual data stream monitoring, 8th international conference on pervasive technologies related to assistive environments, July 2015, Corfu, Greece
15. Anagnostopoulos C, Tsounis A, Hadjiefthymiades S (2007) Context awareness in mobile computing environments. Wirel Pers Commun 42(3):445–464
16. Liu C, Cao G (2010) Distributed monitoring and aggregation in wireless sensor networks. INFOCOM
17. Lozano J, Suarez JI, Arroyo P, Ordiales JM, Alvarez F (2012) Wireless sensor network for indoor air quality monitoring. Chem Eng Trans 30
18. Mainwaring A, Polastre J, Szewczyk R, Culler D, Anderson J (2002) Wireless sensor networks for habitat monitoring. WSNA
19. Anagnostopoulos C, Hadjiefthymiades S (2008) Enhancing Situation-Aware systems through imprecise reasoning. IEEE Trans Mob Comput 7(10):1153–1168
20. Mendel JM (2007) Type-2 fuzzy sets and systems: an overview. IEEE Comput Intell Mag 2(2)
21. Mendel JM (2001) Uncertain Rule-Based fuzzy logic systems: introduction and new directions. Upper Saddle River, Prentice-Hall
22. Ramadan AB, El-Garhy A, Zaky F, Hefnawi M (2012) New environmental prediction using fuzzy logic and neural networks. Int J Comput Sci Issues 9(3)
23. Rothenpelier P, Kruger D, Pfisterer D, Fischer S (2009) FleGSens–Secure area monitoring using wireless sensor networks, int. conference on sensor networks, information and ubiquitous computing
24. Anagnostopoulos C, Hadjiefthymiades S (2009) Advanced inference in situation-aware computing. Trans Sys Man Cyber . Part A 39(5):1108–1115
25. Schneider R, Breedlove D (2001) Fire management study unit, technical report, USDA forest service & Georgia forestry commission, Georgia USA
26. Sharma A, Golubchik L, Govidnan R (2007) On the prevalence of sensor faults in real world deployments, SECON '07, 2007., on the prevalence of sensor faults in real world deployments SECON '07
27. Bottou L (2010) Large–scale machine learning with stochastic gradient descent@. In: Proceedings of the 19th COMPSTAT 2010. Springer, pp 177–187
28. Peskir G, Shiryaev A (2006) Optimal stopping and free-boundary problems, 123–142. Birkhauser, Basel
29. Carpenter G, Grossberg S (1988) The ART of adaptive pattern recognition by a self–organizing neural network. IEEE Computer 21(3):77–88
30. Bertsekas D (2000) Dynamic programming and optimal control (2nd ed.). Athena Scientific
31. Silva T, Zhao L (2012) Stochastic competitive learning in complex networks. IEEE Transactions on Neural Networks and Learning Systems 23(3):385–398
32. Gupta P, Srinivasan R (2011) Missing data prediction and forecasting for water quantity data, proc. intl conf. on modeling simulation and control IPCSIT, vol 10. IACSIT Press, Singapore
33. Newton SC, Pemmaraju S, Mitra S (1992) Adaptive fuzzy leader clustering of complex data sets in pattern recognition. IEEE Trans Neural Networks 3(5):794–800
34. Hinton G, Sejnowski TJ (1986) Learning and relearning in Boltzmann machines, parallel distributed processing, vol 1. MIT Press, pp 282–317
35. Shell J, Coupland S, Goodyer E (2010) Fuzzy data fusion for fault detection in wireless sensor networks. UK Workshop on Computational Intelligence
36. He T, Krishnamurthy S, Stankovic JA, Abdelzaher T, Luo L, Stoleru R, Yan T, Gu L, Hui J, Krogh B (2004) Energy–efficient surveillance system using wireless sensor networks. In: Proceedings of the 2nd international conference on mobile systems, applications, and services (MobiSys '04). ACM, New York, NY, USA, pp 270–283
37. Younis O, Fahmy S (2004) HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans Mob Comput 3(4)
38. Zheng Y, Liu F, Hsieh H-P (2013) U-Air: when urban air quality inference meets big data. In: Proceedings of the KDD, Chicago, Illinois USA
39. Chow YS, Robbins HE, Siegmund D (1971) Great expectations: the theory of optimal stopping. Houghton Mifflin, Boston (Mass.)
40. Kapinas VM, Mihos SK, Karagiannidis G (2009) On the monotonicity of the generalized Marcum and Nuttall Q-functions. IEEE Trans Inf Theory 55(8):3701–3710
41. Abramowitz M (1974) Handbook of mathematical functions, with formulas, graphs and mathematical tables. Dover Publications, Incorporated
42. Shnidman D (1989) The calculation of the probability of detection and the generalized Marcum Q-function. IEEE Trans Inf Theory 35(2):389–400

Dr. Christos Anagnostopoulos is an Academic Research Fellow (tenure track) in the School of Computing Science at the University of Glasgow. He is a member of the Information, Data, and Analysis (IDA) and associate member of the Glasgow Systems Section (GLASS) at Glasgow. He is co-founding director of the IDEAS (Information, Data, Events Analytics at Scale) and associate member of NETLAB (Networked Systems Research Laboratory). Dr. Anagnostopoulos is an author of over 100 publications in referred scientific journals/conferences in the areas of context-aware/mobile computing, stochastic optimization, and large-scale machine and statistical learning exploring distributed analytics in Mobile and Edge Computing. He has received funding in excess of £0.5m for his research by the EC, National Initiatives, and the industry (incl. Repado). Dr. Anagnostopoulos before joining Glasgow was appointed as an Assistant Professor at Ionian University and Adjunct Assistant Professor at the University of Thessaly in the area of network-centric information 2 systems. He has held postdoctoral research positions at University of Glasgow (UK/EPSRC) and University of Athens (ECfunded projects) in the areas of large-scale statistical learning & predictive analytics in distributed environments. He holds a BSc (Hons. and Valedictorian) in Informatics & Telecommunications, MSc (distinction) in Advanced Information Systems, and a PhD in Computing Science, University of Athens. He is an associate fellow of the HEA, member of ACM and IEEE.

Kostas Kolomvatsos received his B.Sc. in Informatics from the Department of Informatics at the Athens University of Economics and Business in 1995, his M.Sc. in Computer Science - New Technologies in Informatics and Telecommunications and his Ph.D. from the Department of Informatics and Telecommunications at the National and Kapodistrian University of Athens (UoA) in 2005 and in the beginning of 2013 respectively. He is now a Senior Researcher in the National and Kapodistrian University of Athens - Department of Informatics and Telecommunications. He has participated in several European and national research projects. He is a member of the Pervasice Computing Research Group. His research interests are in the definition of Intelligent Systems and techniques adopting Machine Learning, Computational Intelligence and Soft Computing for Pervasive Computing, Distributed Systems, Big Data and Cloud Computing.