Convey Intelligence to Edge Aggregation Analytics

Natascha Harth, Kostas Delakouridis and Christos Anagnostopoulos

Abstract In Internet of Things (IoT) environments, networks of sensors, actuators, and computing devices are responsible to locally process contextual data, reason and collaboratively support aggregation analytics tasks. We rest on the edge computing paradigm where pushing processing and inference to the edge of the IoT network allows the complexity of analytics to be distributed into many smaller and more manageable pieces and to be physically located at the source of the contextual information it needs to work on. This enables a huge amount of rich contextual data to be processed in real time that would be prohibitively complex and costly to deliver on a traditional centralized cloud/back-end processing system. We propose a lightweight, distributed, predictive intelligence mechanism that supports communication efficient aggregation analytics within the edge network. Our idea is based on the capability of the edge nodes to perform sensing and locally determine (through prediction) whether to disseminate contextual data in the edge network or to locally re-construct undelivered contextual data in light of minimizing the required communication interaction at the expense of accurate analytics tasks. Based on this decision making, we eliminate data transfer at the edge of the network, thus saving network resources for sensing and receiving data, by exploiting the nature of the captured contextual data. We provide comprehensive experimental evaluation of the proposed mechanism over a real contextual dataset and show the benefits stemmed from its adoption in edge computing environments.

Keywords Edge analytics · Aggregation analytics · Predictive intelligence · Communication efficiency · Context prediction

N. Harth · C. Anagnostopoulos (✉)
School of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK
e-mail: christos.anagnostopoulos@glasgow.ac.uk

N. Harth
e-mail: n.harth.1@research.gla.ac.uk

K. Delakouridis
R&D Repado, B. Georgiou & Souri, Athens, Greece
e-mail: kostas.delakouridis@repado.com

R.R. Yager and J. Pascual Espada (eds.), *New Advances in the Internet of Things*,
Studies in Computational Intelligence 715, DOI 10.1007/978-3-319-58190-3_2

1 Introduction

Edge analytics is an approach to efficient contextual data collection and analysis in which computation is performed on sensing devices (sensors, actuators), network switches or other devices (concentrators) instead of transmitting the whole data to a centralized computing environment, e.g., the cloud. Edge analytics has gained attention as the Internet of Things (IoT) paradigm of connected *things* (e.g., sensors, actuators, controllers, concentrators) has become more prevalent [1]. The *edge* of the IoT is where the action is. In IoT environments, high data rate sensors, e.g., video camera, environmental sensors, smart meters are becoming ubiquitous. Today, most high data volumes obtained from such sensors are stored *close* to the point of capture, and only a few are transferred to the cloud. In future, sending all the data from billions of IoT devices to the cloud can overwhelm the existing infrastructure. To overcome these issues, Edge Computing (EC) [2, 3] is emerging bringing contextual data processing, networking, and analytics closer to the IoT devices and applications.

EC represents a shift in which *intelligence* is pushed from the cloud to the edge, localizing certain kinds of analysis, e.g., aggregation operators over data streams and local decision-making [4]. This enables quicker response times, unencumbered by network latency, as well as reduced traffic, by *intelligently processing and relaying the appropriate analyzed data to the cloud*. A primary benefit of edge analytics is scalability. Pushing analytics algorithms to IoT devices alleviates the processing strain on enterprise data management and analytics systems, even as the number of connected devices being deployed by organizations—and the amount of data being generated and collected—increases [5]. By 2020, its projected there will be anywhere from 25 to 50 billion *things* connected to the IoT, i.e., up to seven connected things for every person on Earth. We can anticipate that these billions of connected things will generate data volume far in excess of what can easily be processed and analyzed in the cloud dealing with energy constraints (network lifetime), limited bandwidth, and network latency [6]. Unlike the cloud infrastructure, edge network exhibits the following properties: (i) heterogeneous hardware, (ii) unreliable low bandwidth communication network, (iii) limited, on-board energy budget and limited processing power. Moreover, edge analytics algorithms should not rely on any central coordinator and must be fault-tolerant as node/link failures are a common occurrence. Transferring large volume of data to cloud using low-power radio is often unfeasible due to energy and bandwidth limitation.

We envisage the *edge* of a network as a site for off-loading bandwidth and energy hungry sensor data. To generate value out of the large volume of data on the edge, we need energy-efficient, communication-efficient, autonomous, and lightweight contextual information processing algorithms. We abstract an edge network architecture through *edge nodes* forming a layer between *sensing/actuator nodes* and the cloud. Several Sensing and Actuator Nodes (SAN) are connected to each Edge Node (EN), e.g., cloudlet, sink node, powerful smartphone. Since ENs are located close to the SANs, contextual data should be intelligently transferred to them in real-time and in an energy efficient manner. Each SAN performs measurements and locally determines whether to transfer these measurements to the ENs or not in light of minimizing the required communication interaction (overhead) at the expense of accurate analytics tasks performed on the ENs. Based on this context, our idea rests on locally predict *whether* to disseminate sensed data or not within an edge network to **achieve quality analytics by being energy efficient**. In other words, we attempt to eliminate data transfer at the edge of the network, thus saving network resources for sensing and receiving data, by exploiting the nature of the captured contextual data. However, this comes at the expense of the quality of analytics tasks.

The fundamental requirement to materialize such predictive intelligence at the edge network is (i) the autonomous nature of SANs to locally perform sensing and disseminate data under analytics quality-driven rules and (ii) the capability of the ENs to locally perform lightweight and robust analytics tasks over the data received from their connected SANs. Our major goal is to examine the impact of this predictive intelligence on the quality of the aggregation analytics tasks on the ENs in light of extending the edge network lifetime and being communication efficient.

2 Overview and Motivation

2.1 Aggregation Analytics

All pieces of context captured by SANs and ENs in IoT environments, in general, **contextual information** sources are considered as continuous data streams, where analytics tasks are applied to extract statistical dependencies, aggregate analytics tasks, and infer new knowledge. Context-aware applications, crowd-sensing applications in IoT [7, 8], environmental and geophysical monitoring [9], e.g., forest monitoring [10–13] (through unnamed vehicles), agriculture monitoring [14], road traffic monitoring, surveillance, video analytics [15], marine environment monitoring [16], watershed monitoring systems [17, 18] and statistical analytics applications over large-scale data streams require efficient, accurate and timely data analysis in order to facilitate (near) real-time decision-making, data stream mining, and situational context awareness in IoT environments. The IoT computing devices revolutionize sensing in a wide range of application domains because of their reliability, accuracy, flexibility, cost effectiveness and ease of deployment. A contextual data stream contains values from contextual parameters corresponding to sources (e.g., humidity and temperature sensors of a smart-phones). A set of sources (e.g., mobile sensing and computing devices) captures contextual information representing e.g., a geographically monitored area or the condition of a road network. Context-aware and IoT applications exploit all such context, for instance, to (i) obtain the most affected areas by a fire in the last twenty minutes, (ii) identify a concept drift in certain parameters from eleven o'clock in the morning up to now by applying aggregation functions overs contextual values (e.g., `SUM`, `AVG`), (iii) infer the top-k recent congested segments of city road networks, or (iv) obtain regularly the highest pollution level within a time horizon in a smart city. Many critical IoT applications have been developed on top of contextual data streams captured by IoT devices for events identification. 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 criteria are satisfied [19].

2.2 Overview and Literature Review

A baseline approach for materializing analytics tasks on the cloud is simply all IoT devices to transmit the contextual data from all sensing nodes to certain sink nodes, or back-end system. This has been realized in many previous studies [20–22]. In this case, analytics tasks are carried out by the back-end system on the cloud only, and not by the SANs or ENs at the edge of the network, despite their increasing computing capacity. Evidently, this solution, while practical, has many disadvantages, such as a high energy consumption incurred by transmitting the raw data to the cloud, the need for wireless link bandwidth, and high latency [3].

In the era of EC, instead, the desiderata are:

- *push* the analytics tasks close to the contextual data sources, i.e., to the ENs and
- *push* intelligence to SANs and ENs to *collaboratively* support edge analytics. ENs have to intelligently communicate with the SANs in an energy-efficient way, since communication efficiency is crucial to the prolonged lifetime of the edge network to support edge analytics.

We have distinguished two basic methods for edge analytics. The first method is based on the observation that the SANs and ENs, capable of local computation and sensing, create the possibility of analyzing and building (training) analytics models in a distributed way. In this class of edge analytics, e.g., [23–25], contextual data and/or model's meta-data are circulated within the edge network, which evidently requires energy for data and meta-data dissemination adding extra communication overhead. The second method refers to a group-based communication and single localized computation/processing scheme e.g., [13, 20, 26–30]. In this approach, an EN is responsible for a group of SANs and maintains a set of historical contextual data of each SAN within the group. Such localized method is communication efficient due to the reduced length of routing path from SANs to the cloud. To support such type of edge analytics, energy is consumed on communication, i.e., sending and receiving data from SANs to the EN, and computation, i.e., ENs are processing local data. However, since the cost of local processing and analytics tasks is nontrivial, we should take into account the trade-off between intra-edge-network communication and localized computation [31].

Both above mentioned basic methods are required to be efficient to support edge analytics in terms of computation and communication. The computational efficiency of the analytics tasks is a challenging research area, where recently distributed and large-scale statistical and machine learning algorithms emerge, e.g., [32]; this is beyond the scope of this paper. In the edge network communication aspect, we elaborate on the mechanism of the *selective data delivery*, which is adopted in many distributed computing and sensors environments [31]. We argue that this mechanism can be adopted on EC environments and support communication-efficient edge analytics. Specifically, this mechanism relies on the principle of bounded-loss approximation: a network node locally decides on delivering its sensed data to a network node based on local predictions of representative contextual data, which approximate the actual data given an approximation error bound. The intuition behind this decision is that if a sensed value x on a sensing node is *close* to the predicted value $\hat{x}$ (predicted locally on the sensing node), there is not much benefit by delivering x to the node dedicating for further processing, i.e., the EN in our case. Otherwise, it is important for the EN to consider x to proceed with accurate analytics tasks. Evidently, there is a trade-off, that we should pay attention, between contextual data communication and accuracy of analytics due to approximation. On the one hand, by selectively sending and receiving contextual data within an edge network increases the network life time and the available bandwidth, since less data are circulated. On the other hand, this comes at the expense of quality of the analytics tasks, due to local processing with *deliberately* approximated data at the EN. This requires that:

- The EN employs a mechanism to re-construct the undelivered data in light of proceeding with processing and analytics tasks.
- The SAN is equipped with computation capacity for real-time prediction of $\hat{x}$. This requirement is already provided by EC environments, where IoT devices are armed with both: sensing and computing capabilities.

In this paper we propose an intelligent mechanism that makes use of *all* available resources and capabilities in the edge network to support edge analytics.

2.3 Contribution and Organization

The principle of selective data delivery for supporting edge analytics is not directly adopted. Though, it should be adjusted to *split* the predictive intelligence to the SAN and the EN: the former node *locally predicts* the expected data and *locally decides* on their delivery given an approximation error bound; the latter node *locally predicts/reconstructs* the undelivered data in case that the SAN decided not to deliver. Through this intelligence split, we introduce a mechanism to support all the above mentioned edge-analytics methodologies. That is, such predictive intelligence is applied before a SAN needs to communicate with an EN for an analytics task, while the EN should re-construct the un-delivered data before proceeding with the scheduled analytics task. Should the IoT applications tolerate some *error* in the derived analytics task, like prediction accuracy and quality of aggregation and data fusion operators, then this mechanism can be communication efficient hired for edge analytics, as will be shown in our performance Sect. 5. We show that our mechanism can provide aggregation analytics tasks in the edge network. The fundamental question that we pose in this work is: *Is predictive intelligence on SANs and ENs efficient by trading quality of aggregation analytics for communication saving?* In this paper **we study the impact of this mechanism on edge analytics** in terms of efficiency by sacrificing quality of analytics results. The aim is to achieve a significant increase in the edge network lifetime by tolerating quality of the analytics tasks.

To the best of our knowledge this is a first mechanism that explores the potentials of predictive intelligence on the EC paradigm under the concept of analytics to the edge. The key contributions in this paper are:

- We present a decentralized predictive intelligence mechanism within an edge network with SANs and ENs for supporting edge analytics in a communication efficient way;
- The proposed mechanism is evaluated for showcasing the trade-off between accuracy (quality) of edge aggregation analytics operators and communication overhead;
- We propose certain light-weight re-construction policies for ENs with $O(d)$ computational complexity in a d-dimensional data space using exponential smoothing as a selective data delivery mechanism.
- We evaluate the mechanism using real contextual data from sensors and actuators networks.

The paper is organized as follows: In Sect. 3 we present our rationale and basic concept of the edge predictive intelligence formulated by certain definitions, preliminaries, and the fundamental metrics for evaluating the proposed mechanism. Section 4 reports on the predictive intelligence split to the SAN and EN perspectives elaborating on certain policies for data delivery and re-construction. In Sect. 5 we showcase the performance of the proposed mechanism with a real contextual dataset, while Sect. 6 concludes the paper with future research agenda on edge analytics.

3 Edge Predictive Intelligence

3.1 Rationale

We consider an edge network with connected ENs forming an arbitrary topology. Each EN j is connected with n_j SANs in a tree-like topology with root the EN and leaves its SANs as shown in Fig. 1. A SAN i is connected with its unique EN j and $\mathcal{N}_j = \{1, \ldots, n_j\}$ denotes the SAN set of the EN j, i.e., $i \in \mathcal{N}_j$.

A SAN i at every time instance $t = 1, 2, \ldots$ senses a d-dimensional row vector $\mathbf{x}_t = [x_{1t}, \ldots, x_{dt}] \in \mathbb{R}^d$ of contextual parameters, like temperature, humidity, sound, wind speed, air pollutant chemical compounds, etc. Hereinafter, we refer to $\mathbf{x}$ as *context vector*. The SAN i can communicate with its EN j in the edge network by

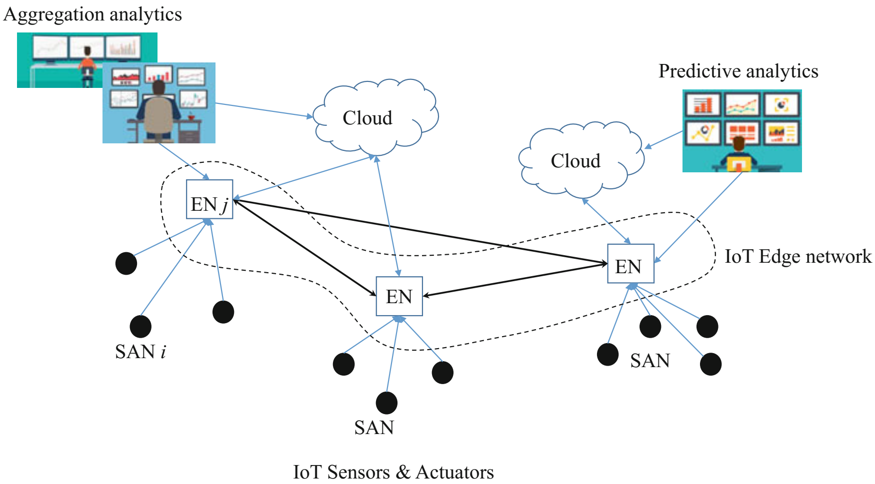


Fig. 1 The edge network with the ENs and the corresponding SANs provide communication efficient analytics to end-users, analysts, and to IoT applications

transferring context vectors. To materialize the proposed predictive intelligence, the SAN i is equipped with a context vector prediction algorithm $f_i(\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-N})$, which uses the recent past $N \geq 1$ sensed context vectors stored in a sliding window $\mathcal{W}$ of size N to predict the context vector $\hat{\mathbf{x}}_t$ at time instance t. That is:

$$\hat{\mathbf{x}}_t = f_i(\mathbf{x}_{t-1}, \ldots, \mathbf{x}_{t-N}) = f_i(\mathcal{W}) \tag{1}$$

and window $\mathcal{W} = (\mathbf{x}_{t-N}, \ldots, \mathbf{x}_{t-1})$. The SAN i, after actually sensing the context vector $\mathbf{x}_t$ at time t, locally predicts the predicted context vector $\hat{\mathbf{x}}_t$, thus, the **local prediction error** is:

$$e_t = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\| \tag{2}$$

where $\|\mathbf{x}\| = (\sum_{k=1}^{d} x_k^2)^{1/2}$ is the Euclidean norm of $\mathbf{x}$. Such prediction capability yields the SAN able to decide whether to send context vectors $\mathbf{x}$ to its EN j or not for further processing. SAN i relies on a θ-based context vector delivery decision rule:

- **Case 1** If the predicted $\hat{\mathbf{x}}_t$ differs from the actual sensed $\mathbf{x}_t$ with respect to a *decision threshold* $\theta > 0$, i.e., $e_t > \theta$, then the SAN i sends the actual $\mathbf{x}_t$ to the EN j.
- **Case 2** Otherwise, i.e., $e_t \leq \theta$, the SAN i does not send $\mathbf{x}_t$ to the EN j. In this case, the EN j is responsible for *reconstructing* a context vector locally for further processing.

In Case 1, the EN j receives the transmitted context vector $\mathbf{x}_t$ from SAN i. In Case 2, the EN j is equipped with a re-construction function

$$\bar{\mathbf{x}}_t = g_j(\mathbf{u}_{t-1}, \ldots, \mathbf{u}_{t-M}) = g_j(\mathcal{W}) \tag{3}$$

of the recent $M \geq 1$ context vectors $\mathbf{u}$ from a sliding window $\mathcal{W} = (\mathbf{u}_{t-M}, \ldots, \mathbf{u}_{t-1})$ to locally predict (reconstruct) the undelivered vector $\mathbf{x}_t$, notated by $\bar{\mathbf{x}}_t$ through historical context vectors. Specifically, the context vectors $\mathbf{u}$ in the EN's sliding window $\mathcal{W}_j$ correspond to either the actual received context vectors $\mathbf{x}$ from the SAN i (Case 1) or the past locally re-constructed context vectors $\bar{\mathbf{x}}$ from g_j (Case 2), i.e.,

$$\mathbf{u}_t = \begin{cases} \mathbf{x}_t & \text{if } e_t > \theta \text{ (Case 1)} \\ \bar{\mathbf{x}}_t = g_j(\mathcal{W}) & \text{, otherwise; (Case 2)} \end{cases}$$

The **re-construction difference** at the EN j is then:

$$a_t = \begin{cases} 0 & \text{Case 1,} \\ \|\mathbf{x}_t - \bar{\mathbf{x}}_t\| & \text{Case 2.} \end{cases} \tag{4}$$

The sliding window at SAN i contains only actual (sensed) context vectors $\mathbf{x}$, while the sliding window at EN j contains either actual context vectors $\mathbf{x}$ (received from SAN i) or re-constructed context vectors $\bar{\mathbf{x}}$ locally generated by EN j. The difference (norm) between the predicted context vector $\hat{\mathbf{x}}$ on SAN i and the reconstructed context vector $\bar{\mathbf{x}}$ at EN j is $\|\hat{\mathbf{x}} - \bar{\mathbf{x}}\| = \|\mathbf{e} - \mathbf{a}\|$, with $\mathbf{a} = \bar{\mathbf{x}} - \mathbf{x}$ and $\mathbf{e} = \hat{\mathbf{x}} - \mathbf{x}$. This difference is zero when both the predictor and the re-constructor on SAN i and EN j, respectively, result in the same *error*. Overall, when $e_t > \theta$, the reconstruction difference $a_t = 0$, while when $e_t \leq \theta$, the reconstruction difference $a_t \geq 0$.

Goal: Given a decision threshold $\theta > 0$ at SAN i, we study the performance of certain aggregation analytics tasks on EN j. We qualitatively derive sufficient conditions for this and reveal that the decision is a function of both the desired error bound and the correlation among the sensed contextual data values. When the decision threshold is very tight or the correlation is not significant, the SAN i always has to send its context vectors to the EN j. Due to the characteristics and inherent dynamics of the SANs' contextual data, when the underlying data distribution evolves over time, prediction techniques may not work efficiently for a set of less predictable contextual data. We provide certain definitions and preliminaries before elaborating on our distributed intelligence mechanism.

3.2 Definitions and Problem Formulation

Definition 1 (*Sliding Window*) A sliding window $\mathcal{W}$ is specified by a fixed-size temporal extent $N > 0$ ('horizon') by appending new context vectors and discarding older ones on the basis of their appearance.

For instance, at time t, a sliding window $\mathcal{W}$ is a sequence of all context vectors observed from $t - N$ to $t - 1$, i.e., $\mathcal{W} = (\mathbf{x}_{t-N}, \mathbf{x}_{t-N+1}, \ldots, \mathbf{x}_{t-1})$. As an example, an analytics query over $\mathcal{W}$ could be: 'continuously return all context vectors of the past hour, i.e., $N = 60$ min'. The sliding window is the most widely used in continuous aggregation and fusion analytics functions [33–36].

The **aggregation analytics** tasks are evaluated over the contents of a window $\mathcal{W}$. The aggregated results change over time as the window slides. We use the classification from [37] that divides aggregation functions into three categories: distributive, algebraic, and holistic. Let $\mathcal{W}$, $\mathcal{W}_1$, and $\mathcal{W}_2$ be windows. An aggregation analytics function $h : \mathcal{W} \to \mathbb{R}^d$ is distributive if $h(\mathcal{W}_1 \cup \mathcal{W}_2)$ can be computed from $h(\mathcal{W}_1)$ and $h(\mathcal{W}_1)$ for all $\mathcal{W}_1$, $\mathcal{W}_2$. An aggregation analytics function h is algebraic if there exists a 'synopsis function' σ such that for all $\mathcal{W}$, $\mathcal{W}_1$, and $\mathcal{W}_2$: (1) $h(\mathcal{W})$ can be computed from $\sigma(\mathcal{W})$; (2) $\sigma(\mathcal{W})$ can be stored in constant memory; and (3) $\sigma(\mathcal{W}_1 \cup \mathcal{W}_2)$ can be computed from $\sigma(\mathcal{W}_1)$ and $\sigma(\mathcal{W}_2)$. An aggregation analytics function h is holistic if it is not algebraic. Among the standard aggregates, `MAX` and `MIN` are distributive, `AVG` is algebraic, since it can be computed from a synopsis containing `SUM` and `COUNT`, and `QUANTILE`, `MEDIAN` are holistic.

Example 1 We can define the `AVG` and `MAX` analytics functions: $h^{avg}(\mathcal{W}) = \frac{1}{N} \sum_{k=t-N}^{t} \mathbf{x}_k$ and $h^{max}(\mathcal{W}) = [\max\{x_{1k}\}, \ldots, \max\{x_{dk}\}]_{k=t-N}^{t}$, respectively.

In our case, the aggregation analytics function h is running on EN j *for each* sliding window $\mathcal{W}$ containing M received and/or re-constructed context vectors from the SAN $i \in \mathcal{N}_j$ depending on Case 1 and Case 2. Note that such functions are built-in constructs in IoT-application specific continuous analytics queries.

Example 2 The aggregation analytics query 'every minute find the average temperature and the maximum humidity over context streams 'temperature' and 'humidity' collected during the past hour' in Continuous Query Language [38] involving `AVG` and `MAX` operators in a sliding window $\mathcal{W}, N = 60$ min can be expressed as follows:

```
SELECT AVG(temperature), MAX(humidity)
FROM Context Streams [RANGE 60 MINUTES SLIDE 1 MINUTE]
```

Note, typical progressive aggregates like `SUM`, `MIN` and `AVG` requires constant time $O(1)$ per value since there is no need to scan the entire window [39, 40]. More advanced aggregation analytics functions like outliers detection or concept drift detection in a sliding window $\mathcal{W}$ require multiple scanning of the $\mathcal{W}$. Aggregation analytics functions can be also combined on a EN to infer certain events that might trigger decision making.

Example 3 Consider the evaluation of a situational context (localized event stream processing) for the past ten minutes as the activation of the following rule with conjunctive predicates associated with `AVG` and `MAX` aggregation analytics functions over 'temperature' and 'wind-speed' sliding windows from two corresponding SANs:

```
EVENT := IF AVG(temperature) ≥ 90 AND MAX(wind-speed) ∈
[10,20]
WITHIN 10 minutes THEN ACTION is 'warning'
```

Definition 2 (*Aggregation Analytics Difference*) Consider an EN j and its SAN $i \in \mathcal{N}_j$. The aggregation analytics difference β_i between the analytics result on EN j derived from aggregation function h over the window $\mathcal{W}$ in the EN j and the *actual* analytics result derived from h over the window $\mathcal{W}^*$, which contains only the actual context vectors from SAN i to EN j (ground truth) is:

$$\beta_i = \|h(\mathcal{W}) - h(\mathcal{W}^*)\|. \quad (5)$$

The aggregation analytics difference β_i denotes how much the aggregation results over the window $\mathcal{W}$ on EN j with context vectors $\mathbf{u}$ *differ* from the aggregation results over the window $\mathcal{W}^*$ with context vectors $\mathbf{x}$, should SAN i have sent all context vectors to EN j. Obviously, if we encounter only the Case 1, then $\beta_i = 0, \forall i \in \mathcal{N}_j$. Now, since we allow SAN i to decide on sensing context vectors w.r.t. θ and EN j being able to re-construct undelivered context vectors, then $\beta_i \geq 0$. The concept is how much an IoT application tolerates this difference in analytics results in light of communication efficiency in the edge network.

Hence, given a decision threshold $\theta > 0$, our aim is to examine the impact of our predictive intelligence mechanism on (i) the re-construction difference a in (4) and (ii) the aggregation analytics difference β in light of communication efficiency by saving significant network bandwidth.

4 Predictive Intelligence Split

The intelligence of the proposed mechanism is split into two parts: (i) the SAN's intelligence with respect to the local prediction algorithm f_i and (ii) the EN's intelligence with respect to the local re-construction algorithm g_j that supports the analytics tasks introduced in Sect. 3.2.

4.1 Sensor-Actuator Node Intelligent Part

Consider a SAN i and let us elaborate on the first part. Very complex prediction models are not practical in the discussed EC paradigm due to the limited (energy-constrained) computational capacity of the SANs. Fortunately, simple linear predictors are sufficient to capture the temporal correlation of realistic contextual data as shown by previous studies [41–43]. A sliding window-based linear predictor is one of popular approaches to predicting the future based on past N measurements.

In this work, we are seeking to reduce the computational power for prediction and to use a small fraction of the SAN's computing power by adopting a predictive function with low complexity and computational effort. Multivariate exponential smoothing, used for time series forecast, is an ideal predictor adopted in our case, as its computational complexity is $O(d)$ in a d-dimensional space (explained later). A simple exponential smoothing weighs the current sensed context vector $\mathbf{x}_t$ and the historic context vectors [44]. This simple smoothing function is adopted as the prediction function f_i for the θ-based decision making.[1]

At each time t, a smoothed context vector $\mathbf{s}_t$ is calculated by using the current sensed context vector $\mathbf{x}_t$ and the previous smoothed vector $\mathbf{s}_{t-1}$, i.e.,

$$\mathbf{s}_t = \alpha \mathbf{x}_t + (1 - \alpha)\mathbf{s}_{t-1} \tag{6}$$

initializing with $\mathbf{s}_0 = \mathbf{x}_0$. The relationship between the history of the measured data and the current data is represented by $\alpha \in [0, 1]$. A higher α denotes more importance to the current values and less importance to the historic values. Normally, $\alpha = 0.7$ [44]. The calculated smoothed vector $\mathbf{s}_{t-1} = [s_{1,t-1}, \dots, s_{d,t-1}]$ refers to the predicted context vector $\hat{\mathbf{x}}_t$, that is: $\hat{\mathbf{x}}_t = f_i(\mathcal{W}_i) = \mathbf{s}_{t-1}$, with the window $\mathcal{W} = (\mathbf{s}_{t-1})$ at SAN i containing only the recent smoothed context vector. Hence, the complexity of f_i is $O(d)$; we require d computations for smoothing the $\mathbf{s}_t$ in (6) at time instance t. The forwarding decision of the actual $\mathbf{x}_t$ to the EN j depends whether the prediction error $e_t = \|\mathbf{s}_{t-1} - \mathbf{x}_t\|$ exceeds the threshold θ.

4.2 Edge Node Intelligent Part

On the other side, the EN j, at time instance t either receives $\mathbf{x}_t$ (Case 1) or nothing (Case 2). In Case 1, the EN j simply inserts the delivered $\mathbf{x}_t$ into its corresponding window $\mathcal{W}$ (which is associated with the SAN $i \in \mathcal{N}_j$) discarding the oldest context vector, i.e., $\mathbf{u}_t = \mathbf{x}_t$. In Case 2, the EN j encounters an undelivered vector problem, since there is nothing to push in the sliding window $\mathcal{W}$. Such undelivered context vectors must be re-constructed with the available context vectors $\mathbf{u}$ reside currently in the $\mathcal{W}$ at EN j. In order to achieve this, we propose three re-construction policies,

[1] Double exponential smoothing (Holt-Winters time series smoothing) could be adopted dealing with the same computational complexity.

i.e., variants of the re-construction function $g_j(\mathcal{W})$. We should stress that, we require a computationally efficient re-construction function on EN j, thus, being relatively a small overhead compared to the analytics tasks. Those policies are discussed below.

Policy 1 This policy, in Case 2, uses the *most* recent context vector from $\mathcal{W}$ at EN j, i.e., the first element of the sliding window, as the re-constructed context vector. Therefore, the re-constructed context vector is inserted into the $\mathcal{W}$ and the oldest context vector from the window is discarded. Note that, after this insertion, there are two duplicates of the most recent context vector in the window. There might be also the case where the entire window (of length N) would have contained the same context vector if the SAN i had not sent a context vector in the last N time instances. This denotes that, during this recent history of N time instances, the maximum difference of the sequentially sensed context vectors measured on SAN i is less that θ. In this case, it is redundant to send *similar* context vectors to EN j given a threshold θ. In Case 1, the EN j simply inserts the delivered $\mathbf{x}_t$ into the window and discards the oldest context vector.

Policy 2 This policy, in Case 2, re-constructs the undelivered context vector $\bar{\mathbf{x}}_t$ as the average vector of the current context vectors in the window $\mathcal{W}$, i.e.,

$$\bar{\mathbf{x}}_t = g_j(\mathcal{W}) = \frac{1}{N} \sum_{k=t-N}^{t-1} \mathbf{u}_k.$$

This re-constructed value is then inserted into the window discarding the oldest one. In Case 1, the EN j simply inserts the delivered $\mathbf{x}_t$ into the window and discards the oldest context vector.

Policy 3 This policy applies the exponential smoothing algorithm (discussed above) for re-constructing the undelivered context vector in the EN j. In Case 1, the EN j simply inserts the delivered $\mathbf{x}_t$ into the window and discards the oldest context vector. Moreover, after this insertion, the EN j calculates the smoothing context vector $\mathbf{s}'_t$ based on the delivered $\mathbf{x}_t$ and the previously calculated smoothed context vector, i.e.,

$$\mathbf{s}'_t = \alpha \mathbf{x}_t + (1 - \alpha)\mathbf{s}'_{t-1}.$$

In Case 2, this policy re-constructs the $\bar{\mathbf{x}}_t$ with the recently smoothed context vector $\mathbf{s}'_{t-1}$ (exploiting the context vector delivery in Case 1) and discards the oldest context vector from the window. Note that, the series of the smoothed vectors $\mathbf{s}'_t$ in EN j is not the same with the series of the smoothed vectors $\mathbf{s}_t$ in SAN, since the vectors $\mathbf{s}'_{t-1}, \mathbf{s}'_{t-2}, \ldots$ are calculated by the $\mathbf{u}_{t-1}, \mathbf{u}_{t-2}, \ldots$ vectors from the window $\mathcal{W}_i$ on EN j. Moreover, in Case 2, after the re-construction of $\bar{\mathbf{x}}_t$ with $\mathbf{s}'_{t-1}$, the smoothed context vector for time instance t is $\mathbf{s}'_t = \alpha \bar{\mathbf{x}}_t + (1 - \alpha)\mathbf{s}'_{t-1} = \mathbf{s}'_{t-1}$. Overall, Policy 3 at the EN j has as follows:

$$\begin{cases} \mathbf{s}'_t = \alpha \mathbf{x}_t + (1 - \alpha)\mathbf{s}'_{t-1} & \text{, Case 1} \\ \bar{\mathbf{x}}_t = \mathbf{s}'_{t-1} \text{ and } \mathbf{s}'_t = \mathbf{s}'_{t-1} & \text{, Case 2.} \end{cases} \tag{7}$$

5 Performance Evaluation

5.1 Dataset and Experimental Setup

In our experiments, we used a real dataset for assessing the performance of the proposed edge prediction intelligence mechanism. The contextual dataset (DS1) was adopted by UCI [45]. This dataset contains twelve SANs of chemical compounds and environmental parameters: CO, PT08.S1 (tin oxide), Non Metanic HydroCarbons, Benzene, PT08.S2 (titania), NOx, PT08.S3 (tungsten oxide), NO_2, PT08.S4, PT08.S5 (indium oxide), temperature, relative humidity, and absolute humidity. All these contextual parameters are required to measure the air pollution of a specific area. These data are collected every hour and refer to $T = 9357$ multi dimensional measurements ($d = 12$) with $n = 12$ SANs and one EN. Inside this dataset missing values occur. For each SAN, we impute those missing values by adopting the missing value imputation method of linear interpolation. This method exploits two data points (x_0, y_0) and (x_1, y_1) to reconstruct a linear function to find for a specific x value the missing value y as follows: $y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0)$.

For comparison and reproduction, DS1 is *normalised* and *scaled*, i.e., each contextual parameter $x \in \mathbb{R}$ is mapped to $\frac{x-\mu}{\sigma}$ with mean value μ and variance σ^2, and scaled in [0,1] using $\frac{\max\{x\}-x}{\max\{x\}-\min\{x\}}$.

For sensitivity analysis of our mechanism, the experiments were carried out with different values of the decision threshold $\theta \in \{10^{-5}, 10^{-3}, 10^{-2}, 0.05, 0.06\}$. Using a normalised and scaled dataset, θ is interpreted as the percentage change of a measured/sensed value x by: 0.0002%, 0.02%, 2%, 10% and 12% respectively for the chosen θ values, respectively. The chosen θ values are examined for the impact of the local prediction error e_t in (2) on our mechanism to decide on forwarding a measurement. The local predictor/exponential smoother in the SAN side, in our experiment, uses $\alpha \in \{0.5, 0.7, 1\}$ as suggested in [44]. Moreover, in Policy 3, the reconstruction smoother function in EN side uses $\alpha = 0.7$. The sliding window size is set to $N = 10$. This selected size represents for DS1 a history of the last 10 hours.

Overall, our experimental set up includes five θ values and three α values over three different policies (Policy 1, 2, and 3) for reconstruction on the EN. This leads to an overall of 75 experiments for each of the aggregation analytics function $h(\mathcal{W})$: i.e., `AVG`, `MAX` and `MIN` analytics function for evaluation. In order to objectively assess the performance of our mechanism, we implemented the baseline mechanism. This mechanism is produced by capturing the continuous contextual data and transmitting them from all SANs to an EN, without any predictive intelligence on SAN or EN.

5.2 *Performance Metrics*

We firstly define the performance metrics of **counter of communications**, i.e., the number of sensed values are sent from a SAN i to its EN j. By adopting the baseline mechanism the number of communications is allocated with $12 \cdot 9357 = 112,284$. To better illustrate and compare our mechanism with the baseline, the counter of communications is indicated as 100%. Using our mechanism, the counter is only increasing if θ is exceed and values are transmitted from the SANs to the EN. The overall communication of n SANs over T sensing values, in our method is then:

$$c(T) = \sum_{t=1}^{T} \sum_{i=1}^{n} I_{i,t}, \tag{8}$$

with $I_{i,t} = 1$ if SAN i sends its sensed value to the EN; otherwise $I_{i,t} = 0$. Evidently, the overall communication of n SANs over T sensing values, in the baseline method is $T \cdot n$, since $I_{i,t} = 1, \forall i, t$. The percentage of communication is then $\frac{c(T)}{Tn}$.

The re-construction difference a in (4), and the aggregation analytics difference β in (5) is evaluated using the **symmetric mean absolute percentage error (SMAPE)**. During the experiments, we calculated the average SMAPE per SAN for each time instance $t \in \{1, \ldots, T\}$. This metric is used to represent a percentage error of SMAPE $\in [0, 100]$ because of its unbiased properties [46]. SMAPE is defined for reconstruction difference a and aggregation analytics difference β as:

$$\text{SMAPE} = \begin{cases} \frac{100}{T} \sum_{t=1}^{T} \frac{a_t}{\|\mathbf{x}_t\| + \|\bar{\mathbf{x}}_t\|} & , \|\mathbf{x}_t\| + \|\bar{\mathbf{x}}_t\| > 0 \text{ for } a, \\ \frac{100}{T} \sum_{t=1}^{T} \frac{\beta_t}{\|h(\mathcal{W})\| + \|h(\mathcal{W}^*)\|} & , \|h(\mathcal{W})\| + \|h(\mathcal{W}^*)\| > 0 \text{ for } \beta. \end{cases} \tag{9}$$

Our major aim is to demonstrate the trade-off between communication and re-construction difference/aggregation analytics difference. That is, we can tolerate some *slight* increase in the analytics error by gaining a *significant decrease* in the number of communications, thus, being communication efficient and prolonging the edge network lifetime. It is evaluated how the aggregated analytics functions $h(\mathcal{W})$ and re-construction inside an EN behave with decreasing the number of communications towards the EN. This decrease of communications is produced by increasing the value of θ and changing the value of α inside the SAN (exponential smoother).

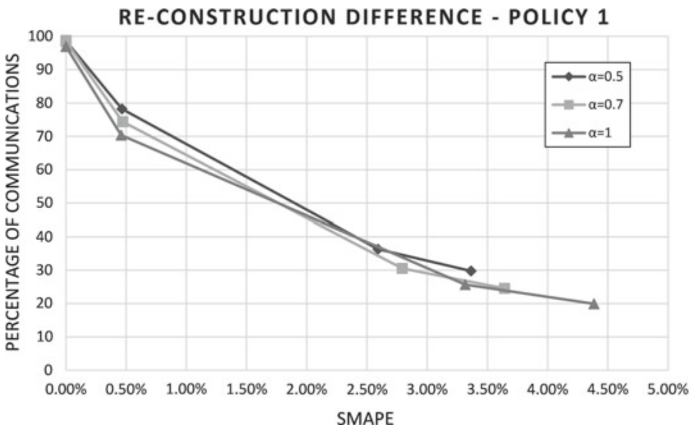


Fig. 2 Re-construction Difference trade-off for Policy 1; percentage of communication against SMAPE

5.3 *Performance Assessment*

Performing our method of predictive intelligence with the explained chosen values, we can illustrate the trade-off between communication and error for the re-construction and aggregation analytics differences. Independent of the specific differences, our aim is to reduce the percentage of communication only with a slight increase of the error.

Generally, increasing the value of θ is decreasing the number of communications towards the EN. The reason behind this is that θ is demonstrating the tolerance for a change of the expected value and the actual/sensed one. Therefore high values of θ are indicating that values can vary between a larger range before they are sent towards the EN. Furthermore, it is notable that the number of communications is highly dependent on the exponential smoother parameter α. Given the same θ value, the number of communications is decreasing with higher values for α. Increasing α means reducing the influence of previous/historical measured data and weight the current data high. Having a value for $\alpha = 1$, the current measured value is compared against only the previous for a forwarding decision.

5.3.1 Re-Construction Difference Assessment

Evaluating the re-construction difference for DS1 it is applicable that the chosen reconstruction policy inside the EN is highly influencing the produced SMAPE. Given the three different policies, Policy 2 is generating the highest error values over $\alpha \in \{0.5, 0.7, 1\}$ and $\theta \in \{10^{-5}, 10^{-3}, 10^{-2}, 0.05, 0.06\}$. Policy 3 as reconstruction method is creating nearly the same error percentage as Policy 1. Policy 1 represents the best possible solution for reconstruction the undelivered values inside the EN.

Comparing Policy 1 against different values for α it can be observed from Fig. 2, that increasing α is increasing the SMAPE. Considering the trade-off between SMAPE and percentage of communication, higher values for α illustrate a lower SMAPE with less communication. Figure 2 shows the re-construction difference for Policy 1 over different α values. His worth noting that, as shown in Fig. 2, using our proposed predictive intelligence for edge analytics on DS1, a communication overhead of 30% can be saved by tolerating an re-construction error of less than 1%. If IoT applications can tolerate up to 2% error in their analytics accuracy, it is possible to save between 50 and 60% of communication. Saving this amount of communication with a given tolerated error would increase the lifetime of an edge network between 30 and 50%.

5.3.2 Aggregation Analytics Assessment

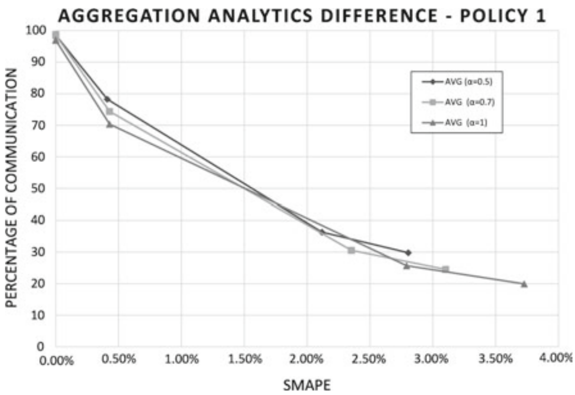
Besides the re-construction difference, the aggregation analytics difference produced by our proposed method is an important metric by many analytics IoT applications. From our experiments it is applicable for all three aggregation functions, `AVG`, `MIN` and `MAX`, that Policy 2 is producing the highest SMAPE for the aggregation analytics difference. Similar to the re-construction difference, Policy 1 is generating the lowest average error per SAN over the entire time frame T. In our experiments a value of $\alpha = 0.5$ is producing the lowest error over all three aggregation functions by employing the re-construction Policy 3. Comparing Policy 1 over all three aggregation functions, Fig. 3 shows this for DS1. Specifically, one can observe from Fig. 3 that, similar to the re-construction difference, the aggregation analytics difference depends on α. Higher values of α producing a better trade-off. For `MIN` and `MAX` this reverses after a SMAPE of around 1.5%. `AVG` continuously produces a good trade-off for high α.

For DS1 it is observed in Fig. 3, that a reduction of 20% by communication for `MIN`, `MAX` and 30% for `AVG` is only generating an error of 0.5%. Therefore, it is possible to increase the lifetime of a network up to 30% with tolerating a slightly difference towards the true result. For applications that could tolerate a higher discrepancy for this kind of aggregation functions, they can save up to 50% with an error of 1.5–2%.

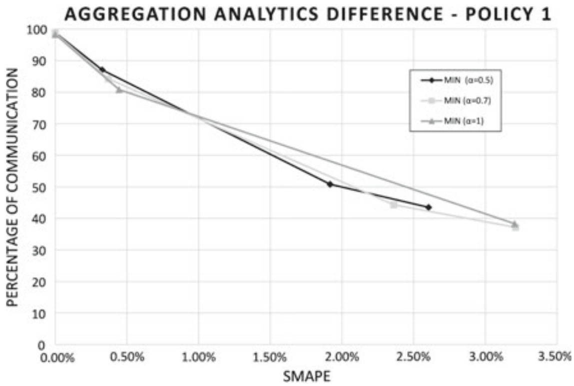
6 Conclusions

We focus on the edge computing paradigm where pushing aggregation analytics to the edge of the IoT network allows the complexity of analytics tasks to be distributed into many smaller and more manageable pieces and to be physically located at the source of the contextual information it needs to work on. We introduce a lightweight, distributed, predictive intelligence mechanism that supports communication efficient aggregation analytics within the edge network of SANs and ENs. The mechanism

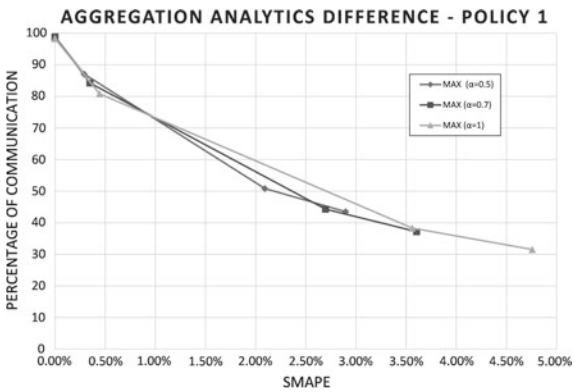
Fig. 3 Aggregation Analytics Difference trade-off for DS1 with Policy 1; percentage of communication against SMAPE



(a) AVG function



(b) MIN function



(c) MAX function

is based on the idea of locally deciding whether to deliver contextual data or not in light of minimizing the induced communication overhead and providing high quality analytics tasks. Based on splitting this intelligence into: prediction (through exponential smoothing) and decision making at the SANs and context re-construction at ENs (by proposing three policies), we eliminate data transfer at the edge of the network, by exploiting the predictability of the captured contextual data. We provide comprehensive experimental evaluation of the proposed mechanism over a real multidimensional contextual dataset for aggregation analytics tasks and show the benefits stemmed from its adoption in edge computing environments. We experimented with the trade-off between accuracy (quality) of edge analytics tasks and communication overhead. Our mechanism demonstrated its efficiency in supporting high quality of edge analytics by tolerating a relatively low error in light of decreasing significantly the communication overhead in an edge network. Our future agenda includes certain modifications of our mechanism to support predictive analytics tasks including outliers detection and linear regression predictive models in edge computing environments.

References

1. M. Satyanarayanan et al., Edge Analytics in the Internet of Things, in *IEEE Pervasive Computing,* vol. 14, no. 2. pp. 24–31, Apr-June 2015
2. The mobile-edge computing initiative, http://www.etsi.org/technologies-clusters/technologies/mobile-edge-computing
3. I. Stojmenovic, S. Wen, The Fog computing paradigm: scenarios and security issues, in *2014 Federated Conference on Computer Science and Information Systems, Warsaw*, 2014, pp. 1–8
4. S. Yi, C. Li, Q. Li, A survey of fog computing: concepts, applications and issues, in *Proceedings of the 2015 Workshop on Mobile Big Data*, 2015, pp. 37–42
5. A. Vulimiri, C. Curino, P.B. Godfrey, T. Jungblut et al., WANalytics: geo-distributed analytics for a data intensive world, in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, 2015, pp. 1087–1092
6. B. Cheng, A. Papageorgiou, M. Bauer, Geelytics: enabling on-demand edge analytics over scoped data sources, in *IEEE International Congress on Big Data (BigData Congress)*. San Francisco, CA, vol. 2016, pp. 101–108, 2016
7. R. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges. Commun. Mag. IEEE **49**(11), 32–39 (2011)
8. N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, A. Campbell, A survey of mobile phone sensing. Commun. Mag. IEEE **48**(9), 140–150 (2010)
9. L.M. Oliveira, J.J. Rodrigues, Wireless sensor networks: a survey on environmental monitoring. J. Commun. **6**(2), 143–151 (2011)
10. J. Kang et al. High-fidelity environmental monitoring using wireless sensor networks, *Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys '13)* (USA, Article 67, 2013)
11. A. Awang et al. RIMBAMON: a forest monitoring system using wireless sensor networks, in *ICIAS 2007*, pp. 1101–1106, 2007
12. E. Zervas et al., Multisensor data fusion for fire detection. Inf Fusion Elsevier **12**(3), 1566–2535 (2011)
13. C. Anagnostopoulos, S. Hadjiefthymiades, K. Kolomvatsos, Accurate, dynamic, and distributed localization of phenomena for mobile sensor networks. ACM Trans. Sen. Netw. **12**, 2, Article 9 (April 2016), 59 pages (2016)

14. S. Nittel, A Survey of geosensor networks: advances in dynamic environmental monitoring. Sensors **9**, 5664–5678 (2009)
15. Yu. Pieter Simoens, P. Pillai Xiao, Z. Chen, K. Ha, M. Satyanarayanan, Scalable crowdsourcing of video from mobile devices, in *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '13)* (ACM, New York, NY, USA, 2013), pp. 139–152
16. G. Xu et al., Applications of wireless sensor networks in marine environment monitoring: a survey. Sensors **14**(9), 16932–16954 (2014)
17. G.W. Eidson et al., The south carolina digital Watershed: end-to-end support for realtime management of water resources, in *Proceedings of the 4th International Symposium on Innovations and Real-time Applications of Distributed Sensor Networks (IRADSN 09)*, vol. 2010 (USA, May 2009)
18. N. Nguyen et al. A real-time control using wireless sensor network for intelligent energy management system in buildings, in *Proceedings of the IEEE Worskshop on Environmental Energy and Structural Monitoring Systems (EESMS 10)*, pp. 87–92, Sept 2010
19. K. Kolomvatsos; C. Anagnostopoulos; S. Hadjiefthymiades, Data fusion and type-2 fuzzy inference in contextual data stream monitoring, in *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. PP, no. 99, pp. 1–15, June 2016
20. S.M. McConnell, D.B. Skillicorn, A distributed approach for prediction in sensor networks, in *Proceedings of the SIAM International Conference on Data Mining Workshop Sensor Networks*, 2005
21. D. Tulone, S. Madden, An energy-efficient querying framework in sensor networks for detecting node similarities, in *Proceedings of the IEEE/ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM)*, 2006
22. S. Goel, T. Imielinski, Prediction-based monitoring in sensor networks: taking lessons from MPEG. ACM SIGCOMM Comput. Commun. Rev. **31**(5), 82–98 (2001)
23. A. Simonetto, G. Leus, Distributed maximum likelihood sensor network localization, in *IEEE Transactions on Signal Processing*, vol. 62, no. 6, pp. 1424–1437, 15 Mar 2014
24. G. Kejela, R.M. Esteves, C. Rong, *Predictive Analytics of Sensor Data Using Distributed Machine Learning Techniques*, pp. 626–631, 2014
25. R. Gemulla, E. Nijkamp, P.J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '11)* (ACM, New York, NY, USA, 2011), pp. 69–77
26. C. Anagnostopoulos, S. Hadjiefthymiades, Advanced principal component-based compression schemes for wireless sensor networks. ACM Trans. Sen. Netw. **11**, 1, Article 7, 34 pages (2014)
27. C. Anagnostopoulos, S. Hadjiefthymiades, A. Katsikis, I. Maglogiannis, Autoregressive energy-efficient context forwarding in wireless sensor networks for pervasive healthcare systems. Pers. Ubiquitous Comput. **18**(1), 101–114 (2014)
28. C. Anagnostopoulos, S. Hadjiefthymiades, P. Georgas, PC3: principal component-based context compression. J. Parallel Distrib. Comput. **72**(2), 155–170 (2012)
29. A. Manjeshwar, D.P. Agrawal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in *Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS '01)* (IEEE Computer Society, Washington, DC, USA), p. 189
30. K. Papithasri, M. Babu, Efficient multihop dual data upload clustering based mobile data collection in Wireless Sensor Network, in *2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore*, pp. 1–6, 2016
31. H. Jiang, S. Jin, C. Wang, Prediction or not? an energy-efficient framework for clustering-based data collection in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **22**(6), 1064–1071 (2011). June
32. L. Bottou, F.E. Curtis, J. Nocedal, Optimization Methods for Large-Scale Machine Learning, arXiv:1606.04838
33. M. Dallachiesa, G. Jacques-Silva, B. Gedik, K.-L. Wu, T. Palpanas, Sliding windows over uncertain data streams. Knowl. Inf. Syst. (2014)

34. K. Patroumpas, T. Sellis, Maintaining consistent results of continuous queries under diverse window specifications. Inf. Syst. **36**(1), 42–61 (2011)
35. D.J. Abadi, D. Carney, U. Cetintemel, M. Cherniack, C. Convey, S. Lee, M. Stonebraker, N. Tatbul, S. Zdonik, Aurora: a new model and architecture for data stream management. VLDB J. **12**(2), 120–139 (2003)
36. D.J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A.S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, S. Zdonik, The design of the Borealis stream processing engine, in *CIDR*, Jan 2005
37. J. Gray, S. Chaudhuri et al., Data cube: a relational aggregation operator generalizing group-by, cross-tab, and sub totals. Data Min. Knowl. Discov. **1**(1), 29–53 (1997). Mar
38. A. Arasu, S. Babu, J. Widom, The CQL continuous query language: semantic foundations and query execution. VLDB J. **15**(2), 121–142 (2006). June
39. K. Patroumpas, T. Sellis, Multi-granular time-based sliding windows over data streams, in *2010 17th International Symposium on Temporal Representation and Reasoning (TIME)*, pp. 146–53, 2010
40. K. Patroumpas, T. Sellis, Window specification over data streams, in *Proceedings of the International Conference on Current Trends in Database Technology (EDBT'06)* (Springer, Berlin, 2006), pp. 445–464
41. D. Chu, A. Deshpande, J.M. Hellerstein, W. Hong, Approximate data collection in sensor networks using probabilistic models, in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 2006
42. V.P. Chowdappa, C. Botella, B. Beferull-Lozano, Distributed clustering algorithm for spatial field reconstruction in wireless sensor networks, in *IEEE 81st vehicular technology conference (VTC Spring), Glasgow*, vol. 2015, pp. 1–6, 2015
43. C. Anagnostopoulos, T. Anagnostopoulos, S. Hadjiefthymiades, An adaptive data forwarding scheme for energy efficiency in Wireless Sensor Networks, in *5th IEEE International Conference Intelligent Systems* (London, 2010), pp. 396–401
44. J. Durbin, S. Jan Koopman, *Time Series Analysis by State Space Methods* (Oxford Statistical Science Series, 2012)
45. S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia, On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. Sens Actuators B: Chem. **129**(2), 750–757, 22 Feb 2008. ISSN 0925-4005
46. C. Tofallis, A better measure of relative prediction accuracy for model selection and model estimation. J. Oper. Res. Soc. **66**(8), 1352–1362

Springer

http://www.springer.com/978-3-319-58189-7

New Advances in the Internet of Things
Yager, R.R.; Pascual Espada, J. (Eds.)
2018, VII, 178 p. 75 illus., 39 illus. in color., Hardcover
ISBN: 978-3-319-58189-7