Real-Time Data Analytics in Sensor Networks

Themis Palpanas[1]
University of Trento
themis@disi.unitn.eu

Abstract. The proliferation of Wireless Sensor Networks (WSNs) in the past decade has provided the bridge between the physical and digital worlds, enabling the monitoring and study of physical phenomena at a granularity and level of detail that was never before possible. In this study, we review the efforts of the research community with respect to two important problems in the context of WSNs: real-time collection of the sensed data, and real-time processing of these data series.

Keywords: Wireless Sensor Networks, Data Collection, Data Analytics, Outliers, Deviation Detection, Uncertain Data Series, Data-Aware Network Protocols

1 Introduction

In the past decade we have witnessed the proliferation of Wireless Sensor Networks (WSNs), fueled by advances in processor technologies and wireless communications that led to the development of small, low cost and power efficient sensor nodes [1–3]. The great benefit they provide is that they serve as the bridge between the physical and digital worlds, and they enable us to monitor and study physical phenomena at a granularity and level of detail that was never before possible.

Collecting the data sensed by the WSN to a centralized server (the sink), or being able to directly query the WSN are probably the most important functionalities that a WSN has to support. Lots of work has been directed to how to efficiently achieve these goals, where the primary objective is to extend the WSN lifetime, while fulfilling the application requirements (collecting the required data, or answering the queries).

There are two main ideas that researchers have explored: first, data are correlated (both across time and over space), and second, several applications accept small errors in the data values they operate on. These ideas have led to the development of a multitude of techniques that trade accuracy for time performance and energy savings.

In this study, we review the efforts of the research community with respect to the problems of real-time collection of the sensed data, and real-time processing of these data series in the context of a WSN. Furthermore, we examine the interplay between such data management techniques and network protocols.

We note that the aim of this study is not an exhaustive enumeration and discussion of all the related works, but rather, the description of prominent research problems that have been studied so far with regards to the sensor data processing and analysis, as well as of promising future research directions.

2 Data Collection

The availability and use of sensor networks have generated a lot of research interest. A major part of this effort has concentrated on how to collect the sensed data at the sink (where they will be further processed and analyzed), using the least amount of energy[1] possible. The challenge arises from the special characteristics of WSNS and the nature of the data they produce, namely: limited resources, intermittent connections, and spatio-temporal correlation of the sensed values [4–6].

Several frameworks for the efficient execution of queries and collection of data in a sensor network have been developed in the last years [4, 7, 8]. The focus in these works was to propose data processing and optimization methods geared specifically toward sensor networks (we describe those in detail later on). The early studies described in-network aggregation techniques for reducing the amount of data transmitted by the nodes, while subsequent research focused on model-driven [9] and data-driven [10] data acquisition techniques. Other works have proposed techniques that take into account missing values, outliers, and intermittent connections [11, 12, 6, 13].

A different approach is based on Kalman filters [14], with the same goal of reducing the required communication among nodes and the sink. Other techniques offer solutions for efficient spatio-temporal data suppression [5, 15–19], where in addition to the temporal correlations present in the sensor network data, they aim at identifying and exploiting the spatial correlations of the data, as well. Furthermore, previous works have proposed algorithms that help in the selection of representative nodes when we want to monitor large-scale phenomena (i.e., phenomena that evolve over days, or months, and involve several sensor nodes) [20], or when we want to take into account the remaining energy of each individual node [21]. The above techniques help to further reduce the communication cost of the sensor network, and could be applied on top of the model-driven, or data-driven techniques.

In the rest of this section, we will discuss techniques in the areas of model-driven and data-driven data acquisition, as well as in spatio-temporal data suppression.

2.1 Model-Driven Data Acquisition

The aim of the model-driven approach is to (conceptually) collect, or process queries on all the data sensed by the WSN, based on probabilistic models that capture the correlations that exist in these data. We note that sensor readings exhibit such correlations in a wide range of domains and applications. This is true, because often times sensors are monitoring slow-changing phenomena with high temporal resolution and/or high spatial resolution. Moreover, correlations may also be present among different types of readings coming from the same sensor node (e.g., it has been shown that temperature and voltage readings are correlated [9]; at the same time it is much less expensive to take voltage readings than temperature).

The model-driven approach works as follows. During an initial training phase, all the sensed data are collected from the nodes in the network, in order to train the probabilistic models that are stored in the sink. Then, these models are used in order to

[1] Given that radio communication in WSNS is much more expensive than CPU processing, this translates to reducing communication and data transfer.

estimate the sensed values, and additionally provide probabilistic guarantees on the correctness of these estimates. Therefore, instead of querying the sensors, we operate on the data produced by the models. If the guarantees produced by the models for these data do not satisfy the accuracy requirements of the application, then we can request additional real data values from the sensors, in order to refine the models to the point that the probabilistic guarantees satisfy the application requirements.

We can now formally define the model-driven data acquisition problem.

Problem 1 (Model-Driven Data Acquisition). Given a sensor network, and a sink that needs to collect all the sensed values within ε of the real value with confidence[2] at least $1-\delta$, design a data collection protocol such that the energy used by the sensor network is minimized.

In order to solve this problem, we need to decide on the probabilistic models to use for approximating the distributions of the sensed values, and also on the communication strategies among the sensors and the sink. Both these aspects of the problem are addressed by the studies that we discuss in the next paragraphs.

Proposed Techniques The BBQ system [9] proposes sensor data acquisition techniques based on time-varying multivariate Gaussian probabilistic models, but other models can alternatively be used, such as probabilistic graphical models [22]. Using the above approach, the produced models capture correlations both among sensed values from the same sensor across time, and among different sensors across space. We note that the above approach requires some knowledge of the special characteristics of the data distribution, such as periodic drifts, which should be encoded in the space of models considered. This means that some minimum amount of domain knowledge is required, in order to make effective use of these techniques.

A similar framework for modeling sensor network data is proposed by Guestrin et al. [23]. The goal is for groups of nodes in the network to collaborate in order to fit a global function to each of their local measurements. This approach employs kernel linear regression in order to model the sensed values, by capturing spatio-temporal correlations. Once again, we observe that this is a parametric approximation technique, and as such, requires the user to make an assumption about the number of estimators required to fit the data. Moreover, there is a need for a training phase (where the models are built, evaluated, and adjusted), which in practice can be rather lengthy and expensive.

Even though the domain knowledge requirement that the above techniques have may be prohibitive for some applications, we note that a large number of applications (where the measured phenomena are known or understood, or when a domain expert is available) can still benefit from such techniques.

2.2 Data-Driven Data Acquisition

The model-driven approach described earlier can lead to significant energy savings for the data acquisition task. However, by the nature of their techniques, they can only pro-

[2] The *confidence* is the probability with which the value recorded by the sink is within ε of the sensed value.

vide probabilistic guarantees on the accuracy of the data that the sink collects, and hence no absolute bound on the error. While this may be sufficient for certain applications (e.g., temperature and humidity monitoring for Heating, Ventilation and Air Conditioning systems), there exists a class of applications, for which hard accuracy guarantees are essential (e.g., scientific applications that need accurate, fine-grained monitoring of some phenomenon).

In several scientific applications, it may also be the case that the domain experts do not already have a model of the data distribution they are sampling using the WSN, but are rather interested in collecting accurate measurements in order to build such a model [24]. Indeed, WSNs offer a unique opportunity to scientists to observe phenomena and develop models for them at a scale and granularity that were never before possible. Nevertheless, in order to so, they need to have accuracy guarantees on the sensor measurements.

In data-driven data acquisition, we make the assumption that the application running at the sink allows for a small tolerance in the accuracy of the reported data. In contrast with the ideal requirements of the sink obtaining *exact* values in *all* data reports, the correctness of these applications is unaffected as long as *i)* the reported values match *closely* the exact ones; *ii)* inaccurate values occur only *occasionally*. In other words, deviations from the exact reports are acceptable, as long as their extent in terms of difference in *value* and *time interval* during which the deviation occurs are small enough. We capture these assumptions, common to many applications, with the following definitions on value tolerance, ε_V, and time tolerance, ε_T (refer to Figure 1). We use the term *error tolerance*, ε_{VT} to refer to both of them together.

Definition 1 (Value Tolerance). *Let V_i be an exact measurement taken at time t_i. The* value tolerance *is defined by the maximum relative and absolute errors acceptable, $\varepsilon_V = (\varepsilon^{rel}, \varepsilon^{abs})$. From the application perspective, reading a value V_i becomes equivalent to reading any value $\hat{V}_i$ in the range R_V defined by the maximum error, $\hat{V}_i \in R_V = [V_i - \varepsilon, V_i + \varepsilon]$, where $\varepsilon = \max\{\frac{V_i}{100}\varepsilon^{rel}, \varepsilon^{abs}\}$. In other words, the application considers a value $\hat{V}_i \in R_V$ as* correct.

Note that the value tolerance includes both an absolute and a relative component. This is useful for applications that involve sensor readings with wide ranges of values.

Definition 2 (Time Tolerance). *Let $T = |t_j - t_k|$ be a time interval, and $\hat{V_T} = \{\hat{V}_j, \ldots, \hat{V}_k\}$ the set of values reported to the application during T. The* time tolerance *ε_T is the maximum acceptable value of T such that* all *the values reported in this interval are incorrect, i.e., $\hat{V}_i \notin R_V$, $\forall \hat{V}_i \in \hat{V_T}$.*

Similarly to the model-driven approach, each node (or group of nodes) in the WSN generates a model for the sensed data. This model is then sent to the sink, along with the last reading. From that point on, the sink can predict the readings of the node based on this shared model. The node is also checking whether its model can accurately describe its own readings (within the error tolerance agreed with the sink), and if this is not true then it computes a new model and transmits it to the sink. Evidently, the sink always records accurate data (i.e., within ε_{VT}), regardless of the quality of the model. The

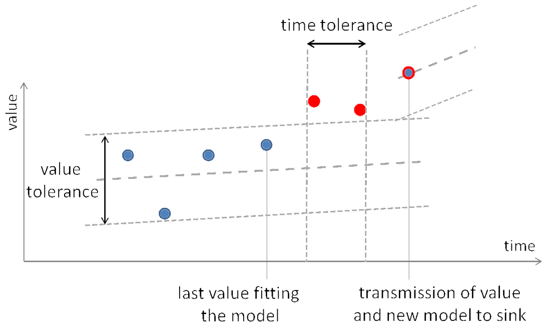


Fig. 1. Value and time tolerance, assuming a linear model (depicted by the thick dashed line) for the sensed data [25].

model quality affects only the effectiveness of the proposed scheme in terms of energy savings.

We can now formally define the problem of data-driven data acquisition.

Problem 2 (Data-Driven Data Acquisition). Given a sensor network, and a sink that needs to collect all the sensed values within ε_{VT}, design a data collection protocol such that the energy used by the sensor network is minimized.

This problem statement is deliberately vague on the specificities of the design of such a protocol. In the following paragraphs we review several techniques that solve this problem, each one focusing on different aspects of the problem. Some studies focus on the selection of the sensed data model (shared among sensors and sink), others concentrate on the effective identification of temporal and/or spatial correlations among the sensed data, while others explicitly aim at maximizing the lifetime of the *entire* sensor network[3].

Proposed Techniques The KEN technique [26] builds and maintains dynamic probabilistic models over the sensor readings, taking into account the spatio-temporal correlations that exist in the sensor readings. These models organize the sensor nodes in non-overlapping groups, and are shared by the sensor nodes and the sink. The expected values of the probabilistic models are the values that are recorded by the sink. If the sensors observe that these values are more than ε_{VT} away from the sensed values, then a model update is triggered.

The PAQ [27] and SAF [28] methods employ linear regression and autoregressive models, respectively, for modeling the measurements produced by the nodes, with SAF leading to a more accurate model than PAQ.

[3] Note that by minimizing the energy consumption of the network, it is possible that the energy of a few specific sensor nodes is depleted much faster than the average. Obviously, this is not desirable, since it may jeopardize the correct operation of the entire network.

Silberstein et al. [29, 10] describe for providing continuous data without continuous reporting, but with checks against the actual data. To achieve this goal, this approach introduces temporal and spatio-temporal suppression schemes, which use the in-network monitoring to reduce the communication rate to the central server. Based on these schemes, data is routed over a chain architecture. At the end of this chain, the nodes that are most near to central server send the aggregate change of the data to it. Since in this scheme (and in data-driven approaches in general) the loss of a model update is crucial[4], special provision is taken for handling network failures [10], so as to ensure correctness.

A recent study proposes a new linear model, DBP [25]. The model is trained using m data points, where the first and the last l points are called *edge points*, and is computed as the slope δ of the segment that connects the average values over the l edge points at the beginning and end of the training phase. This model mitigates the problem of noise and outliers: instead of trying to reduce the approximation error to the *data points* in the recent past, DBP aims at producing models that are consistent with the *trends* in the recently-observed data. Consequently, it leads to improved performance, especially in noisy settings. Moreover, the computation of this model is very simple, and therefore appealing for implementation on resource-scarce nodes.

Another idea that has been studied is to select a set of representative nodes, and use only those for transmitting measurements to the sink. The premise is that each representative node has measurements similar to the measurements of the nodes in its neighborhood. Then, it is only the representative nodes that need to communicate the sensed values to the sink, thus, significantly reducing the energy spent by the WSN.

Data mining approaches contributed to this problem, by providing techniques for clustering and selecting representatives [30–33]. Inside each cluster, the node with the most similar readings to the measurements of all nodes inside that cluster is selected as a cluster representative. Many algorithms were developed to deal with the online distributed clustering of data.

SERENE [34] is a framework for SElecting REpresentatives in a sensor NEtwork. It uses clustering techniques to select the subset of nodes that can best represent the rest of sensors in the network. In order to select an appropriate set of representative sensors, SERENE performs an analysis of the historical readings of sensor nodes, identifies the spatio-temporal correlations among sensors (based on their readings), and groups sensors into clusters according to these correlations. Then, each cluster performs further analysis in order to select the sensors with the highest representation quality. We note that the analysis of the historical data, which has to be repeated when the distribution of the sensor readings changes, may take place in the sensors or in the sink, according to the amount of resources required.

Snapshot Queries [5] is another approach that introduces a platform for energy efficient data collection in sensor networks. By selecting a small set of representative nodes, this approach provides responses to user queries and reduces the energy consumption in the network. In order to select representatives, each sensor node in this

[4] Losing a single model-update message has the potential to introduce large errors at the sink, as the latter will continue to predict sensor values with an out-of-date model until the next one is received.

approach builds a data model of the distribution of measurement values of its neighbors for each attribute. After a node decides which of its neighbors it can effectively represent, it broadcasts its list of candidate cluster members to all its neighbors. Each node selects as its representative the neighbor that can represent it, and that additionally has the longest list of candidate cluster members.

In ECLUN [18], nodes do not continuously communicate with the representatives, but communication is established only when a state change is detected in the monitored phenomena. This communication is further reduced through the careful construction of clusters, which considers similarity in sub-spaces of the full-dimensional sensor readings space. This makes the above approach suitable to deployments of sensor node that produce multi-dimensional readings (i.e., monitor several phenomena simultaneously). ECLUN also tries to uniformly distribute the energy usage among the nodes, resulting in a longer lifetime for the entire sensor network, since the variance of the lifetime of individual nodes is minimized.

A more recent study [35] focuses on the problem of identifying functional dependencies among sensor data streams, in order to determine a small number of sensors from which data are actively collected. The rest of the sensors collect data at lower rates, with the purpose of detecting changes in the discovered dependencies and taking actions to reorganize the sensor data collection process. The dependencies identified in this work are based on regression analysis that takes into account possible lags among the streams.

The above studies use different ways of calculating the correlation among the sensor streams in the network. For this part of the problem, other techniques for identifying correlations in multiple data streams [36–40] could be used as well. The work by Aggarwal et al. [41] describes a method that additionally considers and exploits domain-specific knowledge on the information network of the sensors (i.e., relating to links among the sensors). Another approach for the same problem has proposed a technique for selecting sensors that is based on feedback on the utility of the selected sensors [42].

2.3 Data Series Summarization

Many sensor network applications in diverse domains produce voluminous amounts of data series, such as in meteorology (e.g., temperature measurements [43]), oceanography (e.g., water level measurements [44]), and other domains. The sheer number and size of the data series we need to manipulate in many of the real-world applications mentioned above dictates in several cases the need for a more compact representation of data series than the raw data itself, and a plethora of representations have been proposed to that effect[5].

Even though most data series representations treat every point of the data series equally, there exist WSN applications for which the time position of a point makes

[5] Several techniques have been proposed in the literature for the approximation of data series, including *Discrete Fourier Transform (DFT)* [45, 46], *Discrete Cosine Transform (DCT)*, *Piecewise Aggregate Approximation (PAA)* [47], *Discrete Wavelet Transform (DWT)* [48, 49], *Adaptive Piecewise Constant Approximation (APCA)* [50, 51], *Piecewise Linear Approximation (PLA)* [52], *Piecewise Quadratic Approximation (PQA)* [53], and others. Most of them are amenable to incremental, online operation.

a difference in the fidelity of its approximation. Then we would represent the most recent data with low error, and would be more forgiving of error in older data. We call this kind of time series approximation *amnesic*, since the fidelity of approximation decreases with time, and it therefore requires less memory for the events further in the past (see Figure 2).

For example, the Environmental Observation and Forecasting System[6] [44] operates in a way that allows for some sensors only intermittent connections to the sink (through a repeater station that is not always available). Since the station does not know how long it will be offline, and has a finite buffer, amnesic approximation is an effective way to record the data.

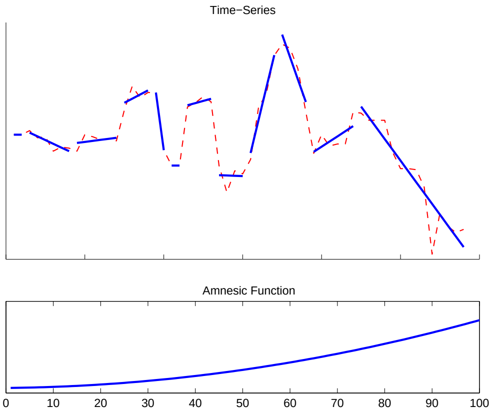


Fig. 2. Depiction of an amnesic approximation, using the piecewise linear approximation technique (the most recent values of the data series are on the left; the oldest values are on the right) [54].

We need a way to specify for each point in time the amount of error allowed for the approximation of the time series. In order to achieve this goal, we use the *amnesic function $A(x)$*, which returns the acceptable approximation error for every point of the data series. We define two forms of amnesic functions, namely, the *relative* and the *absolute* amnesic functions. A relative amnesic function determines the relative approximation error we can tolerate for every point in the time series (e.g., we can specify that when we approximate a point that is twice as old, we will accept twice as much error). When we use relative amnesic functions, we fix the amount of memory that we are allowed to use for the approximation of the data. On the other hand, an absolute amnesic function specifies, for every point in the data series, the *maximum* allowable error for the approximation, which is useful when the application requires quality guarantees for the

[6] This is a large-scale distributed system designed to monitor, model, and forecast wide-area physical processes, such as river systems.

approximation of the data series. When we use absolute amnesic functions, we allow the approximation to use as much memory as necessary in order to meet the error bounds.

More formally, we define the following two problems. More formally, we define the following two problems in the context of *landmark windows*. The *landmark window* is the window that contains all the values of the data series (from a given time point) up to now.

Problem 3 (Landmark Window with Relative Amnesic (URA)). Given a memory budget M and a relative amnesic function $RA(x)$, construct an amnesic approximation using memory at most M that minimizes the approximation error of the data points inside the window.

Problem 4 (Landmark Window with Absolute Amnesic (UAA)). Given an absolute amnesic function $AA(x)$, construct an amnesic approximation that minimizes the required memory M.

Note that in the *URA* and *UAA* problems, the optimization objective is different. In the *URA* problem we seek to minimize the approximation error given the memory space used by the data series approximation, while in the *UAA* problem we want to minimize the space used in the approximation given the maximum error allowed.

Following the definition of the problems for the landmark window, we now define the corresponding problems for the case where we consider the sliding window model.

Problem 5 (Sliding window with Relative Amnesic (SRA)). Given a sliding window W, a memory budget M, and a relative amnesic function $RA(x)$, construct an amnesic approximation using memory M that minimizes the approximation error of the data series within the sliding window.

Problem 6 (Sliding window with Absolute Amnesic (SAA)). Given a sliding window W, and an absolute amnesic function $AA(x)$, construct an amnesic approximation that minimizes the required memory M.

Proposed Techniques Bulut and Singh propose the use of wavelets to represent data streams, which are biased towards the more recent values [55], and describe an efficient, online method for incrementally maintaining this representation. The bias to the most recent values can be seen as a special case of an amnesic function, whose form in this particular case is dictated by the hierarchical nature of the wavelet transform.

A subsequent study [56] generalizes on these ideas, by decoupling the approximation of the time series from a particular dimension-reduction algorithm, and employs user-input to specify how the available memory will be used for the approximation. There has also been relevant work in machine learning, and more specifically, in the neural network community, where the main goal is to model time-varying patterns in data series [57, 58].

A general and efficient solution to the amnesic summarization problems defined earlier is presented in [54]. This study describes solutions for the four variations of the problem, based on online algorithms that use a piecewise linear approximation model.

When a new point arrives, the algorithms update the approximation model in sub-linear time on the number of linear segments.

It has been shown that the techniques mentioned above can be implemented in a very efficient manner in sensor nodes [59]. Moreover, amnesic summarization has been studied in the context of flash memories [60], which offer significant benefits that can be exploited by WSN deployments.

3 Data Processing

Another interesting and important research direction in the context of WSN data management is that of efficient data processing and analysis, and a significant amount of effort has been devoted to it. In this case, we are interested in supporting different types of complex queries in the specific, resource-constrained environment of a WSN.

Several frameworks for the efficient execution of queries in a sensor network have been developed in the past years [4, 7, 8]. The focus in these works was to propose data processing and optimization methods geared specifically towards sensor networks, with the early studies describing in-network aggregation techniques for reducing the amount of data transmitted by the nodes. Ali et al. [61] propose an interesting approach to detect and track discrete phenomena (PDT) in sensor networks. Hellerstein et al. [62] propose algorithms to partition the sensors into *isobars*, i.e., groups of neighboring sensors with approximately equal values during an epoch. Other works have proposed techniques that take into account missing values, outliers, and intermittent connections [11, 12, 6]. We note that some of the techniques we discussed earlier are applicable here (e.g., either to answer adhoc queries [22], or `SELECT*` queries [10]).

In the following paragraphs, we present a framework that enables the development of a variety of complex processing applications in a sensor network. These are applications with high processing requirements over a significant portion of the data generated by the entire WSN. Examples of such applications are the identification and tracking of homogeneous regions, and outlier detection. The identification and tracking of homogeneous regions is used for environmental monitoring (e.g., around oil-drill, or chemical plant sites). In outlier detection, we are interested in discovering exceptional situations that may require the attention of a human analyst: when some of the values of some sensor are not normal, when the number of abnormal values exceeds a given threshold, or when the values of a given sensor are significantly different from the values of its neighbors. We further discuss these applications below.

3.1 Enabling Complex Analytics

The way that streaming applications are able to efficiently process continuous data arriving at high rates, such as those generated by WSNs, is by computing succinct summaries of the data, and operating on these summaries [63, 9].

The framework we describe below aims to approximate in an online fashion multi-dimensional data series distributions [64]. This framework is adaptive and does not require any a priori knowledge about the distributions of the sensed values. Moreover, it operates in a distributed fashion, thus, exploiting all the available resources of the WSN, and reusing any processing that has already taken place.

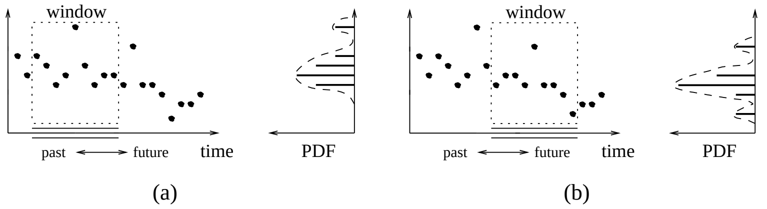


Fig. 3. Estimation of data distribution in sliding window for two time instances (1-d data) [64].

Data Distribution Approximation Framework The proposed framework for estimating the underlying distribution of a streaming data series works both for the sliding time window and the landmark window models [64]. This framework estimates the distribution of the values generated by the sensors using the kernel density estimators [65], which offer the following desirable properties: (i) they are efficient to compute and maintain in a streaming environment; (ii) they can very accurately approximate an unknown data distribution, with no a priori knowledge and (effectively) no parameters; (iii) they can easily be combined and (iv) they scale well in multiple dimensions. The above properties make the framework applicable to large sensor networks, organized in a hierarchical way[7] [69].

In such an online setting, we require that each sensor maintains a model for the distribution of values it generates within a sliding window W (see Figure 3). Such a model can be efficiently and effectively maintained over time. Then, we need to ensure that this mechanism operates in a distributed fashion. Through a model composition mechanism, we are able to take the data distribution models of two (or more) streams, and construct a single model that describes their combined behavior. The framework also proposes mechanisms for incrementally maintaining the models across all levels of the (conceptual) hierarchy, as well as for comparing them in order to determine the similarity of the sensed values. All the above operations can be efficiently supported in real-time by a sensor node [64].

3.2 Detection and Tracking of Homogeneous Regions

The first application is identification and tracking of *homogeneous regions* [61, 62], which are defined as spatial divisions of the field under observation that exhibit similar measured values over time, such as an oil spill detected in the ocean (see Figure 4). The sensors deployed around the origin of the spill can organize themselves into a network and communicate the measurements, to detect regions of varying oil concentrations.

Recent studies propose methods for delineating homogeneous regions by a boundary [71, 72]. However, in several situations we need a more generalized grouping of the sensors, based on the sensed values over a time interval. In general, we would like to

[7] The hierarchical decomposition of the sensor network, as well as the selection of the leaders for each level of the hierarchy, can be achieved using any of the energy-efficient techniques proposed in the literature [66–68].

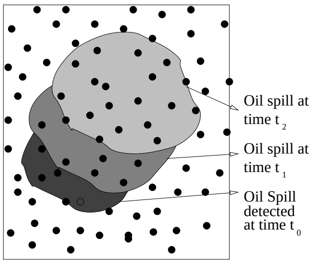


Fig. 4. Spread of an oil spill detected in the water over time [70].

solve the problems of detecting and tracking such homogeneous regions in *real-time* when the definition of the phenomenon is *not* known in advance.

Using the framework described in Section 3.1, we can efficiently identify sensors with similar readings, by comparing their models of the densities of the sensed values [70]. Sensors with very similar models (i.e., data distributions) are grouped together, using the hierarchical organization of the WSN. Each group corresponds to a homogeneous region in space, whose boundaries can be effectively approximated. Then, we can track the movements of these regions over time in a distributed manner, keeping awake only the sensors that are close to the regions that are being tracked. This process is efficiently implemented by tracking the movement of the boundaries of each region.

3.3 Outlier Detection

The second application, which we examine in more detail, is distributed deviation detection in a sensor network. The goal is to identify values (or the corresponding sensor nodes) that look very different from their spatio-temporal neighbors (i.e., the values in the recent history of the sensor stream, or the values in the streams of spatially close sensors). We note that this is a challenging problem, even for static datasets.

This problem is important in a WSN setting because it can be used to identify faulty sensors, and to filter spurious reports from different sensors. Even if we are certain of the quality of measurements reported by the sensors, the identification of outliers provides an efficient way to focus on the interesting events in the sensor network.

In the following subsections, we describe the approaches that have been proposed in the literature, separating them in *approximate* and *exact*, according to whether they provide guarantees on the detection of all the outliers.

Approximate Approaches We first examine outlier identification techniques that cannot provide any hard guarantees on the correctness of the results they produce. Consequently, these techniques may fail to report some of the outliers in the data.

Classification-based

A method based on Bayesian classifiers is described by Elnahrawy et al. [73]. This is a method for modeling and learning statistical contextual information in WSNs, which can also be applied for the task of outlier identification. The employed model assumes that the current reading of each sensor is only influenced by the preceding reading of the same sensor, and the readings of its immediate neighbors. This model is then used to predict the highest probability class of the subsequent reading. If the probability of this class is significantly different from the probability (according to the model) of the actual reading, then this reading is deemed an outlier.

Rajasegarar et al. [74] propose an alternative approach that uses a Support Vector Machine (SVM) classifier. In this case the classification model uses only the information from the past readings of the same sensor node, and ignores the readings from the neighboring nodes.

A drawback of the classification-based approaches is the time and computational effort required in order to train the model that can then be used for outlier detection. This effort can in certain cases be rather high. Note also that for non-stationary data this effort will be continuous.

Data Distribution-based

A technique for outlier detection, based on learning statistical properties of the spatio-temporal correlations of the sensor readings, is proposed by Bettencourt et al. [75]. This technique is geared towards ecological applications, where the sensed pheonomena evolve slowly over time, and are spatio-temporally coherent. According to this technique, sensors learn the distributions of differences among their own readings (over time), as well as the distributions of differences between their readings and the readings of their neighbors. Then, comparing the current readings to these distributions, allows sensors to identify local outliers using a significance test, and a user-specified threshold.

Subramaniam et al. [76] study the case where we wish to identify (among all sensor readings in a sliding window) those values that have very few near neighbors [77], namely, *distance-based* outliers; or those values whose near neighborhood is significantly less dense than their extended neighborhood [78], namely, *density-based* outliers. Note that these definitions do not require any prior knowledge of the underlying data distributions. In order to solve the problem (for both definitions of outliers mentioned above), we need to count the number of sensed values that fall in different regions of the data space. This operation can be efficiently supported by the framework outlined in Section 3.1, and the overall task can be distributed in the entire WSN. Especially for the distance-based outliers, the following observation holds [76]. In a (conceptual) hierarchical organization of the sensor network, a parent node combines in a single pool all the data that its children process. Consequently, outliers have to be identified with respect to this new pool of data. Nevertheless, it is not necessary that the parent node reads in all the data from its childrens input data streams, and for each data value determine whether it is an outlier or not. It suffices for the parent node to examine only the values that have been marked as outliers by its children. All the other data values can be safely ignored, since they cannot possibly be outliers. The above approach allows for the effective distribution of the outlier detection task to the entire WSN, resulting in significant savings in terms of communication messages.

A recent study [79] proposes the use of the hyperellipsoidal model in order to model the normal behavior of sensor nodes. Sensor readings that significant deviate from this model are then declared outliers. The focus of this study is on devising an iterative approach for building and maintaining hyperellipsoidal models, which makes them suitable for non-stationary data distributions.

Node Similarity-based

Zhuang et al. [80] describe an approach for identifying (and cleaning) outliers in a sensor network. They focus on two kinds of outliers: *short simple outliers*, usually represented as an abnormal, sudden burst and depression; and *long segmental outliers*, which represents erroneous sensor readings that last for a certain time period. Their approach works as follows. The Discrete Wavelet Transform (DWT) is applied on the series of sensor readings. The high-frequency coefficients are omitted from the resulting DWT representation, which is subsequently compared to the original data series. Data points that are further away than a distance threshold, d_1, from their DWT representation are deemed short outliers. Then, the data series is compared to the series obtained from other sensors that are geographically close. If no other series is within some distance threshold, d_2, then this data series is deemed a long outlier (similarity between data series is measured using the dynamic time warping distance [81]).

A similar problem is addressed by a subsequent study [82], which targets the identification of outlying sensors. The main observation is that sensors observing the same phenomenon are spatially correlated, but outlying sensor readings are geographically independent. The algorithm described in this study has each sensor compute the difference of its reading to the median reading of its neighboring sensors. Then the sensor collects all these differences from its neighborhood and standardizes them. If the absolute value of its standardized difference is larger than a threshold, d, then this sensor is deemed an outlier.

The TACO framework [83] was recently proposed by Giatrakos et al. to operate in a WSN. In order to identify outliers, TACO takes into account both the history of measurements of a given sensor, as well as the spatial correlations with measurements of other sensors in the vicinity. The outlier detection scheme is based on a two-level hashing mechanism. The first level of hashing takes place locally in each sensor, and is based on Locality Sensitive Hashing [84]. This is used for dimensionality reduction, since the recent history of sensor data readings can be succinctly represented in a space of much smaller dimensionality. Assuming a clustered organization of the sensor network (i.e., hierarchical organization with just two levels), each node communicates this reduced representation of its history to the corresponding cluster-head, which subsequently checks for similar representations among the other nodes in the cluster. Similarity measures such as cosine similarity, Jaccard coefficient and correlation coefficient, are supported. The representations that do not find any similar matches make part of a list of potential outliers that is further communicated to all the cluster-heads of the sensor network. This communication step is efficiently implemented using a second hashing mechanism based on the hamming weight of the representations. Overall, the approach has the advantage that it can provide probabilistic guarantees on the accuracy of the results.

Giatrakos et al. [85] proposed a similar technique, only based on the trends of the sensed data series.

Exact Approaches Unlike the works above, some studies have proposed techniques for outlier detection that guarantee no false negatives (i.e., they identify all outliers). This is a desirable property for several critical applications (e.g., structural integrity monitoring).

The work by Branch et al. [86] describes a technique for distributed outlier detection, where the goal is to identify global outliers (i.e., with respect to the data collected by all sensors). This technique supports definitions of outliers that conform to certain anti-monotonicity and smoothness properties (e.g., it supports the distance to k^{th} nearest neighbor [87], but not the density-based LOF outliers [88]). According to the proposed algorithm, each node maintains a local list of outliers, along with additional information on the data it has transmitted to its neighbors and the data it has received. Following some rounds of peer-to-peer communications, all the nodes in the network converge to the final list of global outliers. This technique guarantees that it will correctly identify all outliers, but only under the assumptions that each node has accurate knowledge of its nearest neighbors, the communications are reliable, and that the data remains static long enough for the algorithm to converge.

In a similar setting, Zhang et al. [89] describe a technique for identification of global outliers, where outliers are defined as the n points with the largest distance to their k^{th} nearest neighbor. This technique assumes the existence of an aggregation tree, which is used as the communication structure among the nodes in the network. The nodes use the aggregation tree to send local outliers and supporting information to their parents, with the root node eventually collecting all the information. At this point the sink is able to calculate the top-n global outlier candidates, which transmits back to all the nodes in the network for verification. If corrections need to be made, these have to be sent to the sink, which will then adjust the candidate outlier list and repeat the verification process. The end result is guaranteed to be correct as long as the network topology does not change, and the algorithm converges to the solution faster than the data gets updated (which implies the need for a rather slow update rate).

A subsequent study [90] takes a more pragmatic approach, removing the assumptions mentioned in the previous approaches. The goal is still to find global outliers. An outlier is defined as a point whose distance from its k^{th} nearest neighbor is more than a distance threshold d; or alternatively, as a point p, such that there exist no more than n other points with distance to their k^{th} nearest neighbors larger than the distance of p to its k^{th} nearest neighbor. This approach is based on the use of an equi-width histogram that can effectively aggregate and summarize the sensor data readings. The histogram is built in the sink, after the sink agrees with all the sensor nodes on the boundaries of the histogram and its buckets. The histogram is then used by the sink in order to prune the search space of outliers, by eliminating all points that cannot possibly be outliers, as well as identifying points that are certainly outliers. For the points for which no definite answer can be given, the sink will explicitly ask the sensor nodes in the network, in an additional round of computations.

Burdakis et al. [91] present an outlier detection framework that can provide hard guarantees on the results. It is based on the Geometric Approach [92], which allows the development of much more efficient methods (in terms of communication cost) than the ones presented above. The Geometric Approach enables the monitoring of complex (potentially non-linear) functions computed over the average of vectors the describe the local behavior at each sensor node, and the handling of different similarity functions (useful for the outlier detection task) in the distributed setting of a WSN: each sensor is assigned a zone, which is locally monitored, and if no sensor identifies a threshold violation in their corresponding zones, then the overall monitored function will not have exceeded the threshold either. Under the proposed framework, we can identify sensor nodes that involve sensed data values (either the recent history of readings, or the vector of the currently sensed values) that are not similar to the corresponding values of other similar nodes in the network. Several different similarity measures can be efficiently supported, including L_1, L_2, L_∞, cosine similarity, extended Jaccard coefficient, and correlation coefficient.

3.4 Processing Uncertain Data Series

In several different domains, such as manufacturing plants and engineering facilities, sensor networks are being deployed to ensure efficiency, product quality and safety [93]: unexpected vibration patterns in production machines, or changes in the composition of chemicals in industrial processes, are used to identify in advance possible failures, suggesting repairs or replacements. However, sensor readings are inherently imprecise because of the noise introduced by the equipment itself [94].

Previous work has shown that treating value uncertainty as a first class citizen can lead to better results in terms of quality and efficiency [93, 95–97]. Since value uncertainty is inherent in WSN data, in the following paragraphs we discuss some recent works on processing data series with uncertain values. The focus of these works is on similarity matching, which serves as the basis for developing various more complex analysis and mining algorithms (e.g., classification, clustering, outlier detection, etc.).

Two main approaches have emerged for modeling uncertain data series. In the first, a Probability Density Function (PDF) over the uncertain values is estimated by using some a priori knowledge [98–100]. In the second, the uncertain data distribution is summarized by repeated measurements (i.e., samples) [101]. We discuss those in more detail below.

Similarity Matching for Uncertain Data Series Formally, an uncertain data series T is defined as a sequence of random variables $< t_1, t_2, ..., t_n >$, where t_i is the random variable modeling the real valued number at timestamp i. All the three models we review and compare fit under this general definition.

The problem of similarity matching has been extensively studied in the past [102–105]: given a user-supplied query sequence, a similarity search returns the most similar data series according to some distance function. More formally, given a collection of data series $C = \{S_1, ..., S_N\}$, where N is the number of data series, we are interested in evaluating the range query function $RQ(Q, C, \varepsilon)$:

$$RQ(Q,C,\varepsilon) = \{S | S \in C \wedge distance(Q,S) \leq \varepsilon\} \quad (1)$$

In the above equation, ε is a user-defined distance threshold. A survey of representation and distance measures for data series can be found elsewhere [106].

A similar problem arises also in the case of uncertain data series, and the problem of probabilistic similarity matching has been introduced in the last years. Formally, given a collection of uncertain data series $C = \{T_1,...,T_N\}$, we are interested in evaluation the probabilistic range query function $PRQ(Q,C,\varepsilon,\tau)$:

$$PRQ(Q,C,\varepsilon,\tau) = \{T|T \in C|Pr(distance(Q,S) \leq \varepsilon) \geq \tau\} \tag{2}$$

In the above equation, ε and τ are the user-defined distance threshold and the probabilistic threshold, respectively.

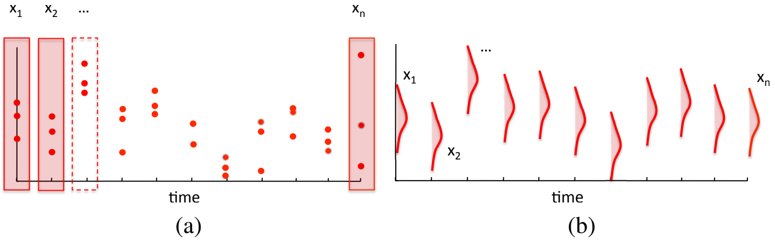
In the recent years three techniques have been proposed to evaluate PRQ queries, namely MUNICH[8] [101], PROUD [99], and DUST [100]. We discuss each one of these three techniques below, and offer some insights in Section 4.2.

Proposed Techniques MUNICH: In [101], uncertainty is modeled by means of repeated observations at each timestamp, as depicted in Figure 5(a). Assuming two uncertain data series, X and Y, MUNICH proceeds as follows. First, the two uncertain sequences X,Y are materialized to all possible certain sequences: $TS_X = \{< v_{11},...,v_{n1} >,...,< v_{1s},...,v_{ns} >\}$ (where v_{ij} is the j-th observation in timestamp i), and similarly for Y with TS_Y. The set of all possible distances between X and Y is then defined as follows:

$$dists(X,Y) = \{L^p(x,y)|x \in TS_X, y \in TS_Y\} \tag{3}$$

The uncertain L^p distance is formulated by means of counting the feasible distances:

$$Pr(distance(X,Y) \leq \varepsilon) = \frac{|\{d \in dists(X,Y)|d \leq \varepsilon\}|}{|dists(X,Y)|} \tag{4}$$



(a) (b)

Fig. 5. Example of an uncertain data series $X = \{x_1,...,x_n\}$ [107], modeled by means of repeated observations (a), and pdf estimation (b).

Once we compute this probability, we can determine the result set of PRQs similarity queries by filtering all uncertain sequences using Equation 4. Note that the naive

[8] We will refer to this method as *MUNICH* (it was not explicitly named in the original paper), since all the authors were affiliated with the University of Munich.

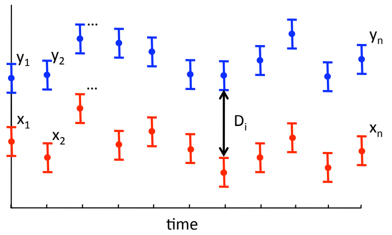


Fig. 6. The probabilistic distance model [107].

computation of the result set is infeasible, because of the exponential computational cost, $|dists(X,Y)| = s_X^n s_X^n$, where s_X, s_Y are the number of samples at each timestamp of X,Y, respectively, and n is the length of the sequences. Efficiency can be ensured by upper and lower bounding the distances, and summarizing the repeated samples using minimal bounding intervals [101]. This framework has been applied to Euclidean and DTW distances and guarantees no false dismissals in the original space.

PROUD: In [99], an approach for processing queries over PRObabilistic Uncertain Data streams (PROUD) is presented. Inspired by the Euclidean distance, the PROUD distance is modeled as the sum of the differences of the streaming data series random variables, where each random variable represents the uncertainty of the value in the corresponding timestamp (see Figure 5(b)). Given two uncertain data series X,Y, their distance is defined as:

$$distance(X,Y) = \sum_i {D_i}^2 \tag{5}$$

where $D_i = (x_i - y_i)$ are random variables, as shown in Figure 6.

According to the central limit theorem, we have that the cumulative distribution of the distances approaches a normal distribution, and the normalized distance follows a standard normal distribution. Therefore, we can obtain the normal distribution of the original distance as follows:

$$distance(X,Y) \propto N(\sum_i E[D_i^2], \sum_i Var[D_i^2]) \tag{6}$$

The interesting result here is that, regardless of the data distribution of the random variables composing the uncertain data series, the cumulative distribution of their distances (1) is defined similarly to their euclidean distance and (2) approaches a normal distribution. Recall that we want to answer PRQs similarity queries. First, given a probability threshold τ and the Cumulative Distribution Function (CDF) of the normal distribution, we compute ε_{limit} such that:

$$Pr(distance(X,Y)_{norm} \leq \varepsilon_{limit}) \geq \tau \tag{7}$$

The CDF of the normal distribution can be formulated in terms of the well known *error-function*, and ε_{limit} can be determined by looking up the statistics tables. Once we have ε_{limit}, we proceed by computing also the normalized ε_{norm}. Then, if a candidate uncertain series Y satisfies the inequality $\varepsilon_{norm}(X,Y) \geq \varepsilon_{limit}$, the following equation holds:

$$Pr(distance(X,Y)_{norm} \leq \varepsilon_{norm}(X,Y)) \geq \tau \tag{8}$$

Therefore, Y can be added to the result set. Otherwise, it is pruned away. This distance formulation is statistically sound and only requires knowledge of the general characteristics of the data distribution, namely, its mean and variance.

DUST: In [100], the authors propose a new distance measure, DUST, that compared to MUNICH, does not depend on the existence of multiple observations and is computationally more efficient. Similarly to [99], DUST is inspired by the Euclidean distance, but works under the assumption that all the data series values follow some specific distribution. Given two uncertain data series X,Y, the distance between two uncertain values x_i, y_i is defined as the distance between their true (unknown) values $r(x_i), r(y_i)$: $dist(x_i,y_i) = L^1(r(x_i),r(y_i))$. This distance can then be used to define a function ϕ that measures the similarity of two uncertain values:

$$\phi(|x_i - y_i|) = Pr(dist(0,|r(x_i) - r(y_i)|) = 0) \tag{9}$$

This basic similarity function is then used inside the dissimilarity function between two uncertain data values x and y, $dust(x,y) = \sqrt{-\log(\phi(|x-y|)) - k}$, where $k = -\log(\phi(0))$, and for entire uncertain sequences takes the following form:

$$DUST(X,Y) = \sqrt{\sum_i dust(x_i,y_i)^2} \tag{10}$$

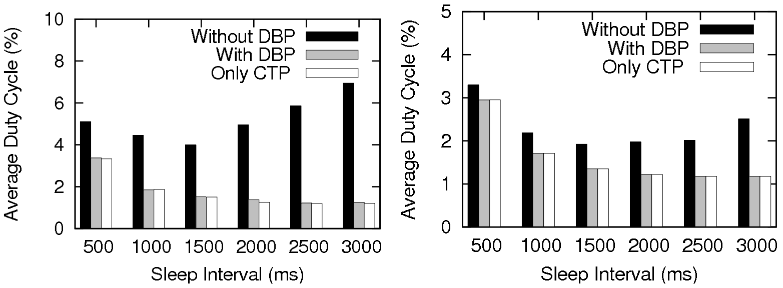
The handling of uncertainty has been isolated inside the ϕ function, and its evaluation requires to know exactly the data distribution. In contrast to the techniques we reviewed earlier, the DUST distance is a real number that measures the dissimilarity between uncertain data series. Thus, it can be used in the place of the existing distance function in mining techniques that have been developed for certain data series.

4 Discussion

In this section, we offer some insights on the approaches and techniques we described earlier. This discussion is also useful for determining promising future research directions.

4.1 Data-Aware Network Protocols

In Section 2, we described several techniques for the efficient collection of the sensed data in a WSN. All these techniques invariably claim considerable savings in terms of required communication messages. Experiments have demonstrated savings of up to $2-3$ orders of magnitude, which is very promising news for the energy savings as well, and consequently the lifetime of the WSN. However, these works have not undertaken a careful study of how the communication savings translate to network lifetime prolongation in real deployments.



(a) Testbed (2 hours).

(b) Tunnel (23 hours).

Fig. 7. Average duty cycle [25]. (Note the difference in the *y*-axis scale.)

A recent study [25] focused on exactly this problem: it examined how DBP (similar results can be obtained for other data-driven data acquisition techniques, as well) affected the WSN lifetime, motivated by a real-world WSN-based application deployment in an operational road tunnel. The performance of DBP was studied in conjunction with the commonly-used network stack composed of CTP [108], BoX-MAC [109], and TinyOS v2.1.1. The experimental evaluation used two settings: an operational road tunnel, and an indoor testbed (fed with the same real data), representative of scenarios with different connectivity. Based on a 47-day, 40-node dataset gathered in this deployment, the study shows that DBP suppresses 99% of the message reports (w.r.t. the baseline, where all nodes send data every 30 sec).

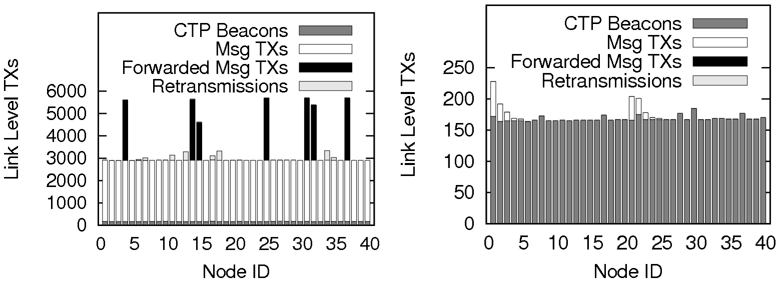
This study examined how data delivery to the application, network lifetime, and routing costs are affected by DBP. To study the impact on lifetime, the study measured the duty cycle of the radio, which is the most power-consuming component. Figure 7 clearly shows that DBP enables significant savings at any sleep interval, while the best sleep interval without DBP is 1500 ms . When using DBP, longer sleep intervals can be used to increase lifetime without affecting data delivery.

Figure 7(a) shows that in the testbed, with a sleep interval of 1500 ms the WSN running DBP lasts twice as long as with no DBP (with the same MAC settings). Using the best sleep interval in both cases (i.e., 1500 and 3000 ms, respectively) yields a three-fold lifetime improvement[9].

A natural question arises at this point: if DBP suppresses over 99% of the messages, why does the network lifetime increase "only" three-fold? This is due to the costs of the network stack, in particular the idle listening and average transmission times of the MAC protocol, and to the overhead of the routing protocol to build and maintain the data collection tree.

To isolate the inherent costs (e.g., tree maintenance) of CTP, experiments were ran with no application traffic. Figure 7 shows the corresponding duty cycle (as *Only CTP*).

[9] The energy savings in the tunnel (see Figure 7(b)) are less remarkable, although still significant, because the network diameter in the tunnel is much smaller w.r.t. the testbed (due to the waveguide effect [110] many direct, 1-hop links to the sink exist, leaving less room for improvement).



(a) Without DBP.

(b) With DBP.

Fig. 8. Tunnel: total link-level transmissions for a sleep interval of 1500 ms [25]. (Note the difference in the *y*-axis scale.)

We observe that the DBP cost is very close to the cost of CTP tree maintenance, regardless of the sleep interval. A finer-grained view is provided by Figure 8, which shows the different components of traffic in the network. Without DBP, the dominate component is message transmission and forwarding; significant retransmissions are present for some nodes, while the component ascribed to CTP (i.e., the beacons probing for link quality) is negligible. When DBP is active, the number of CTP beacons remains basically unchanged. However, because the application-level traffic is dramatically reduced, CTP beacons become the dominant component of the network traffic.

These last observations suggest that further reductions in data traffic would have little practical impact on the system lifetime, as routing costs are dominated by topology maintenance rather than data forwarding. Therefore, improvements are more likely to come from radical changes at the routing and MAC layers: new, data-aware protocols need to be designed, which will take into account the traffic patterns with extremely low data rates that emerge when data-driven data acquisition techniques are employed.

4.2 Uncertain Data Processing

Given the fact that sensors produce values with an inherent uncertainty, and that we are increasingly relying on applications that are driven by sensor data, it becomes evident that efficient and effective processing of uncertain WSN data series is a relevant research direction.

Turning our attention to the three techniques we presented for uncertain data series similarity matching (see Section 3.4), we observe that an important factor for choosing among them is the information that is available about the distribution of the data series and its errors. PROUD requires to know the standard deviation of the uncertainty error and a single observed value for each timestamp, and assumes that the standard deviation of the uncertainty error remains constant across all timestamps. DUST takes as input a single observed value of the data series for each timestamp, and in addition, needs to know the distribution of the uncertainty error for each single time stamp, as well as the distribution of the values of the data series. This means that, in contrast to PROUD, DUST can take into account mixed distributions for the uncertainty errors (albeit, they have to be explicitly provided in the input). MUNICH does not need to know the distribution of the data series values, or the distribution of the value errors: it simply operates on the observations available at each timestamp. When we do not have enough, or accurate information on the distribution of the errors, PROUD and DUST do not offer an advantage in terms of accuracy when compared to Euclidean [107].

All three techniques are based on the simplifying assumption that the values of the data series are independent from one another, which is not true for WSN measurements. A recent study [111] demonstrates that removing this assumption is beneficial: it proposes the UMA and UEMA filters (based on the weighted moving average technique), that in combination with Euclidean distance lead to more accurate results. These results suggest that more work is needed on techniques that take into account the temporal correlations that exist in data series.

The time complexity of these techniques is another important factor. We note that MUNICH is applicable only in the cases where the standard deviation of the error is relatively small, and the length of the data series is also small (otherwise the computational cost is prohibitive), which makes this technique applicable in cases where the sink can do the processing. To a (much) lesser extent, this is also true for PROUD and DUST. On the other hand, UMA and UEMA have significantly lower resource requirements, and could be efficiently implemented in a sensor node.

4.3 Ubiquitous Sensor Networks

Lots of work and research effort has been devoted in the past years in the study of various problems related to WSNs. Several efficient techniques have been developed for the acquisition, management, processing, and analysis of the sensed data, and at the same time (different forms of) WSNs are being deployed in increasingly more domains and situations.

The next frontier in this line of research is the development of very large, ubiquitous WSNs, with increased capabilities for complex, in-network analytics. This vision includes various wireless devices with different specifications (ranging from simple sensor motes to state of the art smartphones), involves advanced, yet efficient, data management and processing techniques, and calls for new breakthroughs in several of the problems and research directions we discussed in the previous sections.

Consider a large WSN deployment, such as *SmartSantander* [112], which comprises of more than $20,000$ sensors in an urban setting. This system has already started to be installed, and can drive the development of powerful applications with a big environmental and societal impact (e.g., environment-aware traffic and transportation monitoring and management, where traffic is managed in real-time, according to levels of pollutants, noise, local events, emergency situations, etc.).

As these WSNs grow larger, covering more space and involving more devices, it makes sense to increase their ability to ingest and process more data in real-time, and to run complex queries in a distributed manner more effectively. This will allow large numbers of queries to run within the WSN, sharing and exchanging results, and with the goal to minimize the need for centralized processing and human intervention (or opportunistically seek human intervention, as in crowdsourcing environments). In order to achieve these goals, methodologies and techniques from other domains could be exploited and adapted (apart from what we have already described here), such as distributed complex event processing [113, 114], and distributed publish/subscribe systems [115].

5 Conclusions

The development of WSN during the past decade helped advance the state of art in several scientific communities that exploited the new opportunities for fine-granularity data-gathering. The popularity of WSNs has also provoked the interest of the research community, and a multitude of studies have been published on techniques and methodologies for the effective and efficient use of the data produced by WSNs, across the networks and data management communities.

As we are now going through the second decade of the WSNs lifetime, we are witnessing a widening and increasing interest in their potential applications, finding their way in new domains and also including new types of devices (e.g., smartphones). In this context, old problems re-emerge, such as the design of novel network protocols that are data-aware, and new challenging problems appear, such as the effective management of uncertainty in sensed data series, and techniques that will scale the in-network complex analytics to very large WSNS.

Acknowledgments

I would like to acknowledge the contributions of my colleagues and collaborators: Alessandro Camerra, Michele Dallachiesa, Adriola Faqolli, Dimitrios Gunopulos, Marwan Hassani, Vana Kalogeraki, Eamonn Keogh, Usman Raza, Katsiaryna Mirylenka, Emmanuel Muller, Amy L. Murphy, Besmira Nushi, Dimitris Papadopoulos, Gian Pietro Picco, Thomas Seidl, Pascal Spaus, Sharmila Subramaniam, Wagner Trupel, Paolo Valleri, and Michail Vlachos. I would also like to thank Charu Aggarwal for his useful comments and suggestions.

References

1. Warneke, B., Last, M., Liebowitz, B., Pister, K.: Smart dust: Communicating with a cubic-millimeter computer. IEEE Computer Magazine (January 2001) 44–51
2. Intanagonwiwat, C., Estrin, D., Govindan, R., Heidemann, J.: Impact of network density on data aggregation in wireless sensor networks. In: ICDCS. (2002)
3. Polastre, J., Szewczyk, R., Culler, D.: Telos: Enabling ultra-low power wireless research. In: Proc. of the Int. Conf. on Information Processing in Sensor Networks (IPSN). (2005)
4. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: Tag: A tiny aggregation service for ad-hoc sensor networks. In: OSDI. (2002)
5. Kotidis, Y.: Snapshot queries: Towards data-centric sensor networks. In: Proceeding of the 21st International Conference on Data Engineering, ICDE. (2005)
6. Wu, W., Lim, H.B., Tan, K.L.: Query-driven data collection and data forwarding in intermittently connected mobile sensor networks. In: Proc. of Int. Conf. on Data Mgmt. for Sensor Networks (DMSN). (2010)

7. Madden, S., Franklin, M.J.: Fjording the stream: An architecture for queries over streaming sensor data. In: Proc. of the Int. Conf. on Data Eng. (ICDE). (2002)
8. Yao, Y., Gehrke, J.: Query processing in sensor networks. In: Proc. of the Conf. on Innovative Data Systems Research (CIDR). (2003)
9. Deshpande, A., Guestrin, C., Madden, S.R., Hellerstein, J.M., Hong, W.: Model-Driven Data Acquisition in Sensor Networks. In: VLDB, Toronto, ON, Canada (2004)
10. Silberstein, A., Filpus, G., Munagala, K., Yang, J.: Data-driven processing in sensor networks. In: Proc. of the Conf. on Innovative Data Systems Research (CIDR). (2007)
11. Gruenwald, L., Sadik, M.S., Shukla, R., Yang, H.: DEMS: a data mining based technique to handle missing data in mobile sensor network applications. In: Proc. of the Int. Conf. on Data Mgmt. for Sensor Networks (DMSN). (2010)
12. Deligiannakis, A., Kotidis, Y., Vassalos, V., Stoumpos, V., Delis, A.: Another outlier bites the dust: Computing meaningful aggregates in sensor networks. In: ICDE. (2009) 988–999
13. Silberstein, A., Gelfand, A.E., Munagala, K., Puggioni, G., Yang, J.: Making sense of suppressions and failures in sensor data: A bayesian approach. In: Proc. of the Int. Conf. on Very Large Data Bases (VLDB). (2007)
14. Jain, A., Chang, E.Y., Wang, Y.F.: Adaptive stream resource management using Kalman filters. In: Proc. of the Int. Conf. on Management of Data (SIGMOD). (2004)
15. Zhou, Z., Das, S., Gupta, H.: Connected k-coverage problem in sensor networks. In: Proc. of the Int. Conf. on Computer Communications and Networks (IC3N). (2004)
16. Vuran, M.C., Akan, O.B., Akyildiz, I.F.: Spatio-temporal correlation: theory and applications for wireless sensor networks. Computer Networks **45**(3) (2004)
17. Jiang, H., Jin, S., Wang, C.: Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks. IEEE Trans. on Parallel Distributed Systems **22** (June 2011)
18. Hassani, M., Müller, E., Spaus, P., Faqolli, A., Palpanas, T., Seidl, T.: Self-organizing energy aware clustering of nodes in sensor networks using relevant attributes. In: Proc. of the Int. Wkshp. on Kowledge Discovery from Sensor Data (SensorKDD). (2010)
19. Pham, N.D., Le, T.D., Choo, H.: SCCS: Spatiotemporal clustering and compressing schemes for efficient data collection applications in WSNs. Int. Journal of Communication Systems **23** (2010)
20. Ali, A., Khelil, A., Shaikh, F.K., Suri, N.: MPM: Map based predictive monitoring for wireless sensor networks. In: Proc. of Int. Conf. on Autonomic Computing and Communication Systems. (2009)
21. Mini, R.A.F., Machado, M.D.V., Loureiro, A.A.F., Nath, B.: Prediction-based energy map for wireless sensor networks. Ad Hoc Networks **3** (2005)
22. Deshpande, A., Guestrin, C., Madden, S.: Using probabilistic models for data management in acquisitional environments. In: CIDR. (2005) 317–328
23. Guestrin, C., Bodik, P., Thibaux, R., Paskin, M., Madden, S.: Distributed Regression: an Efficient Framework for Modeling Sensor Network Data. In: IPSN, Berkeley, CA (2004)
24. Ceriotti, M., Mottola, L., Picco, G.P., Murphy, A.L., Guna, S., Corrà, M., Pozzi, M., Zonta, D., Zanon, P.: Monitoring heritage buildings with wireless sensor networks: The torre aquila deployment. In: IPSN. (2009) 277–288
25. Raza, U., Camerra, A., Murphy, A.L., Palpanas, T., Picco, G.P.: What does model-driven data acquisition really achieve in wireless sensor networks? In: IEEE International Conference on Pervasive Computing and Communications, Lugano, Switzerland (2012)
26. Chu, D., Deshpande, A., Hellerstein, J.M., Hong, W.: Approximate data collection in sensor networks using probabilistic models. In: Proc. of the Int. Conf. on Data Eng. (ICDE). (2006)
27. Tulone, D., Madden, S.: PAQ: Time series forecasting for approximate query answering in sensor networks. In: Proceedings of the European Wkshp. on Wireless Sensor Networks (EWSN). (2006)

28. Tulone, D., Madden, S.: An energy-efficient querying framework in sensor networks for detecting node similarities. In: Proc. of the Int. Conf. on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM). (2006)
29. Silberstein, A., Braynard, R., Yang, J.: Constraint chaining: on energy-efficient continuous monitoring in sensor networks. In: SIGMOD Conference. (2006) 157–168
30. Hassani, M., Müller, E., Seidl, T.: EDISKCO: Energy efficient distributed in-sensor-network k-center clustering with outliers. In: In Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data SensorKDD. (2009) 39–48
31. Rodrigues, P.P., Gama, J., Lopes, L.: Clustering distributed sensor data streams. In: ECML PKDD 2008, LNAI. Springer-Verlag. (2008)
32. Meka, A., Singh, A.K.: Distributed special clustering in sensor networks. In: EDBT 2006, LNCS 3896. (2006) 980–1000
33. Yin, J., Gaber, M.M.: Clustering distributed time series in sensor networks. In: In Proceedings of the Eighth IEEE Conference on Data Mining, ICDM. (2008)
34. Baralis, E., Cerquitelli, T.: Selecting representatives in a sensor network. In: In Proceedings of the SEBD. (2006) 351–360
35. Aggarwal, C.C., Xie, Y., Yu, P.S.: On dynamic data-driven selection of sensor streams. In: KDD. (2011) 1226–1234
36. Yi, B.K., Sidiropoulos, N., Johnson, T., Jagadish, H.V., Faloutsos, C., Biliris, A.: Online data mining for co-evolving time sequences. In: ICDE. (2000) 13–22
37. Zhu, Y., Shasha, D.: Statstream: Statistical monitoring of thousands of data streams in real time. In: VLDB. (2002) 358–369
38. Papadimitriou, S., Sun, J., Faloutsos, C.: Streaming pattern discovery in multiple time-series. In: VLDB. (2005) 697–708
39. Cole, R., Shasha, D., Zhao, X.: Fast window correlations over uncooperative time series. In: KDD. (2005) 743–749
40. Sakurai, Y., Papadimitriou, S., Faloutsos, C.: Braid: Stream mining through group lag correlations. In: SIGMOD Conference. (2005) 599–610
41. Aggarwal, C.C., Bar-Noy, A., Shamoun, S.: On sensor selection in linked information networks. In: DCOSS. (2011) 1–8
42. Golovin, D., Faulkner, M., Krause, A.: Online distributed sensor selection. In: IPSN. (2010) 220–231
43. : Pacific Northwest Weather Data. `http://www-k12.atmos.washington.edu/k12/grayskies/` (2012)
44. Steere, D., Baptista, A., McNamee, D., Pu, C., Walpole, J.: Research Challenges in Environmental Observation and Forecasting Systems. In: Mobile Computing and Networking, Boston, MA, USA (August 2000)
45. Rafiei, D.: On Similarity-Based Queries for Time Series Data. In: International Conference on Data Engineering, Sydney, Australia (March 1999)
46. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast Subsequence Matching in Time-Series Databases. In: ACM SIGMOD International Conference, Minneapolis, MI, USA (May 1994) 419–429
47. Yi, B., Faloutsos, C.: Fast Time Sequence Indexing for Arbitrary LP-Norms. In: VLDB International Conference, Cairo, Egypt (September 2000) 385–394
48. Popivanov, I., Miller, R.J.: Similarity Search Over Time Series Data Using Wavelets. In: International Conference on Data Engineering, San Jose, CA, USA (February 2002) 802–813
49. Chan, K., Fu, W.: Efficient Time Series Matching by Wavelets. In: International Conference on Data Engineering, Sydney, Australia (March 1999) 126–133

50. Chakrabarti, K., Keogh, E.J., Mehrotra, S., Pazzani, M.J.: Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. ACM Transactions on Database Systems **27**(2) (2002) 188–228
51. Lazaridis, I., Mehrotra, S.: Capturing Sensor-Generated Time Series with Quality Guarantees. In: International Conference on Data Engineering, Bangalore, India (March 2003) 429–440
52. Keogh, E.J., Pazzani, M.J.: An Enhanced Representation of Time Series Which Allows Fast and Accurate Classification, Clustering and Relevance Feedback. In: International Conference on Knowledge Discovery and Data Mining, New York, NY, USA (August 1998) 239–243
53. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning. Springer-Verlag (2001)
54. Palpanas, T., Vlachos, M., Keogh, E.J., Gunopulos, D.: Streaming time series summarization using user-defined amnesic functions. IEEE Trans. Knowl. Data Eng. **20**(7) (2008) 992–1006
55. Bulut, A., Singh, A.K.: SWAT: Hierarchical Stream Summarization in Large Networks. In: International Conference on Data Engineering, Bangalore, India (March 2003) 303–314
56. Zhao, Y., Zhang, S.: Generalized dimension-reduction framework for recent-biased time series analysis. IEEE Trans. Knowl. Data Eng. **18**(2) (2006) 231–244
57. Barreto, A., Araujo, A., Kremer, S.: A taxonomy for spatiotemporal connectionist networks revisited: the unsupervised case. Neural Computation **15** (2003) 1255–1320
58. de Vries, B., Principe, J.C.: The gamma model — A new neural model for temporal processing. Neural Networks **5** (1992) 565–576
59. Soroush, E., Wu, K., Pei, J.: Fast and quality-guaranteed data streaming in resource-constrained sensor networks. In: MobiHoc. (2008) 391–400
60. Nath, S.: Energy efficient sensor data logging with amnesic flash storage. In: IPSN. (2009) 157–168
61. Ali, M.H., Mokbel, M.F., Aref, W.G., Kamel, I.: Detection and tracking of discrete phenomena in sensor-network databases. In: SSDBM, Santa Barbara, CA (2005) 163–172
62. Hellerstein, J.M., Hong, W., Madden, S., Stanek, K.: Beyond average: Toward sophisticated sensing with queries. In: IPSN. (2003) 63–79
63. Gilbert, A.C., Kotidis, Y., Muthukrishnan, S., Strauss, M.: Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In: VLDB. (2001) 79–88
64. Palpanas, T., Kalogeraki, V., Gunopulos, D.: Online distribution estimation for streaming data: Framework and applications. In: SEBD. (2007) 430–438
65. Scott, D.: Multivariate Density Estimation: Theory, Practice and Visualization. Wiley & Sons (1992)
66. Ganesan, D., Greenstein, B., Estrin, D., Heidemann, J., Govindan, R.: Multiresolution storage and search in sensor networks. ACM TOS **1**(3) (2005) 27–315
67. Malpani, N., Welch, J., Vaidya, N.: Leader Election Algorithms for Mobile Ad Hoc Networks. In: Proc. Fourth International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications. (2000)
68. Zhao, S., Tepe, K., Seskar, I., Raychaudhuri, D.: Routing protocols for self-organizing hierarchical ad hoc wireless networks. In: IEEE Sarnoff Symposium. (2003)
69. Ye, F., Luo, H., Cheng, J., Lu, S., Zhang, L.: A Two-Tier Data Dissemination Model for Large-Scale Wireless Sensor Networks. In: MOBICOM, Atlanta, GA, USA (2002)
70. Subramaniam, S., Kalogeraki, V., Palpanas, T.: Distributed Real-Time Detection and Tracking of Homogeneous Regions in Sensor Networks. In: RTSS, Rio de Janeiro, Brazil (2006)
71. Chintalapudi, K., Govindan, R.: Localized edge detection in sensor fields. Ad-hoc Networks Journal (2003)

72. Nowak, R., Mitra, U.: Boundary estimation in sensor networks: Theory and methods. In: IPSN, Palo Alto, CA (2003) 80–95
73. Elnahrawy, E., Nath, B.: Context-aware sensors. In: EWSN. (2004) 77–93
74. Rajasegarar, S., Leckie, C., Palaniswami, M., Bezdek, J.C.: Quarter sphere based distributed anomaly detection in wireless sensor networks. In: ICC. (2007) 3864–3869
75. Bettencourt, L.M.A., Hagberg, A.A., Larkey, L.B.: Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks. In: DCOSS. (2007) 223–239
76. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online Outlier Detection in Sensor Data Using Non-Parametric Models. In: VLDB, Seoul, Korea (2006)
77. Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: VLDB, NY, NY (1998)
78. Papadimitriou, S., Kitagawa, H., Gibbons, P., Faloutsos, C.: Loci: Fast outlier detection using the local correlation integral (2003)
79. Moshtaghi, M., Leckie, C., Karunasekera, S., Bezdek, J.C., Rajasegarar, S., Palaniswami, M.: Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks. In: ICDM. (2011) 467–476
80. Zhuang, Y., Chen, L.: In-network outlier cleaning for data collection in sensor networks. In: CleanDB. (2006) 41–48
81. Berndt, D.J., Clifford, J.: Using dynamic time warping to find patterns in time series. In: KDD Workshop. (1994) 359–370
82. Wu, W., Cheng, X., Ding, M., Xing, K., Liu, F., Deng, P.: Localized outlying and boundary data detection in sensor networks. IEEE Trans. Knowl. Data Eng. **19**(8) (2007) 1145–1157
83. Giatrakos, N., Kotidis, Y., Deligiannakis, A., Vassalos, V., Theodoridis, Y.: Taco: tunable approximate computation of outliers in wireless sensor networks. In: SIGMOD Conference. (2010) 279–290
84. Charikar, M.: Similarity estimation techniques from rounding algorithms. In: STOC. (2002) 380–388
85. Giatrakos, N., Kotidis, Y., Deligiannakis, A.: Pao: power-efficient attribution of outliers in wireless sensor networks. In: DMSN. (2010) 33–38
86. Branch, J.W., Szymanski, B.K., Giannella, C., Wolff, R., Kargupta, H.: In-network outlier detection in wireless sensor networks. In: ICDCS. (2006) 51
87. Ramaswamy, S., Rastogi, R., Shim, K.: Efficient algorithms for mining outliers from large data sets. In: SIGMOD Conference. (2000) 427–438
88. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: Lof: Identifying density-based local outliers. In: SIGMOD Conference. (2000) 93–104
89. Zhang, K., Shi, S., Gao, H., Li, J.: Unsupervised outlier detection in sensor networks using aggregation tree. In: ADMA. (2007) 158–169
90. Sheng, B., Li, Q., Mao, W., Jin, W.: Outlier detection in sensor networks. In: MobiHoc. (2007) 219–228
91. Burdakis, S., Deligiannakis, A.: Detecting outliers in sensor networks using the geometric approach. In: ICDE. (2012)
92. Sagy, G., Keren, D., Sharfman, I., Schuster, A.: Distributed threshold querying of general functions by a difference of monotonic representation. PVLDB **4**(2) (2010) 46–57
93. Krishnamurthy, L., Adler, R., Buonadonna, P., Chhabra, J., Flanigan, M., Kushalnagar, N., Nachman, L., Yarvis, M.: Design and deployment of industrial sensor networks: experiences from a semiconductor plant and the north sea. In: Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM (2005) 64–75

94. Ceriotti, M., Corra, M., D'Orazio, L., Doriguzzi, R., Facchin, D., Guna, S., Jesi, G.P., Cigno, R.L., Mottola, L., Murphy, A.L., Pescalli, M., Picco, G.P., Pregnolato, D., Torghele, C.: Is There Light at the Ends of the Tunnel? Wireless Sensor Networks for Adaptive Lighting in Road Tunnels. In: International Conference on Information Processing in Sensor Networks (IPSN). (2011) 187–198
95. Stonebraker, M., Becla, J., DeWitt, D.J., Lim, K.T., Maier, D., Ratzesberger, O., Zdonik, S.B.: Requirements for science data bases and scidb. In: CIDR. (2009)
96. Suciu, D., Connolly, A., Howe, B.: Embracing uncertainty in large-scale computational astrophysics. In: MUD. (2009) 63–77
97. Tran, T.T.L., Peng, L., Li, B., Diao, Y., Liu, A.: Pods: a new model and processing algorithms for uncertain data streams. In: SIGMOD Conference. (2010) 159–170
98. Zhao, Y., Aggarwal, C.C., Yu, P.S.: On wavelet decomposition of uncertain time series data sets. In: CIKM. (2010) 129–138
99. Yeh, M., Wu, K., Yu, P., Chen, M.: PROUD: a probabilistic approach to processing similarity queries over uncertain data streams. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ACM (2009) 684–695
100. Sarangi, S., Murthy, K.: DUST: a generalized notion of similarity between uncertain time series. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM (2010) 383–392
101. Aßfalg, J., Kriegel, H.P., Kröger, P., Renz, M.: Probabilistic similarity search for uncertain time series. In: SSDBM. (2009) 435–443
102. Agrawal, R., Faloutsos, C., Swami, A.: Efficient similarity search in sequence databases. Foundations of Data Organization and Algorithms (1993) 69–84
103. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. ACM SIGMOD Record **23**(2) (1994) 419–429
104. Keogh, E., Chakrabarti, K., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowledge and Information Systems **3**(3) (2001) 263–286
105. Chan, K., Fu, A.: Efficient time series matching by wavelets. In: Data Engineering, 1999. Proceedings., 15th International Conference on, IEEE (2002) 126–133
106. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment **1**(2) (2008) 1542–1552
107. Dallachiesa, M., Nushi, B., Palpanas, T., Mirylenka, K.: Similarity matching for uncertain time series: analytical and experimental comparison. In: QUeST. (2011)
108. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., Levis, P.: The collection tree protocol. In: Proc. of the Int. Conf. on Embedded Networked Sensor Systems (SenSys). (2009)
109. Moss, D., Levis, P.: BoX-MACs: Exploiting Physical and Link Layer Boundaries in Low-Power Networking. Technical Report SING-08-00 (2008)
110. Mottola, L., Picco, G., Ceriotti, M., Guna, S., Murphy, A.: Not All Wireless Sensor Networks Are Created Equal: A Comparative Study On Tunnels. ACM Trans. on Sensor Networks (TOSN) **7**(2) (2010)
111. Dallachiesa, M., Nushi, B., Palpanas, T., Mirylenka, K.: Uncertain time series similarity: Return to the basics. In: VLDB. (2012)
112. : SmartSantander: A City-wide Experimental Facility. `http://www.smartsantander.eu/` (2012)
113. Brenna, L., Gehrke, J., Hong, M., Johansen, D.: Distributed event stream processing with non-deterministic finite automata. In: Proc. of ACM Conference on Distributed Event-Based Systems (DEBS). (2009) 1–12

114. Tanenbaum, A.S., van Steen, M.: 13. In: Distributed Systems: Principles and Paradigms (2nd Edition). Prentice Hall (2006) 603–607

115. Fidler, E., a. Jacobsen, H., Li, G., Mankovski, S.: The padres distributed publish/subscribe system. In: Proc. of the Conference on Feature Interactions in Telecommunications and Software Systems. (2005)