Pattern Detection in Extremely Resource-Constrained Devices

Michael Zoumboulakis and George Roussos

Abstract. Pervasive computing anticipates a future with billions of data producing devices of varying capabilities integrated into everyday objects or deployed in the physical world. In event-based systems, such devices are required to make timely autonomous decisions in response to occurrences, situations or states. Purely decentralised pattern detection in systems that lack time synchronisation, reliable communication links and continuous power remains an active and open research area. We review challenges and solutions for pattern detection in distributed networked sensing systems without a reliable core infrastructure. Specifically, we discuss localised pattern detection in resource-constrained devices that comprise Wireless Sensor and Actuator Networks. We focus on online data mining, statistical and machine learning approaches that aim to augment decentralised pattern detection and illustrate the properties of this new computing paradigm that requires stability and robustness while accommodating severe resource limitations and frequent failures.

1 Introduction

The vast majority of research in middleware and distributed event-based systems proposes techniques that are not directly applicable to the extremely resource-constrained nodes of a Wireless Sensor Network (WSN). This is because they rely heavily on core infrastructure services such as reliable communication links, time synchronisation and a persistent event history. We target a class of pervasive computing devices with constraints in terms of power, processing, memory, bandwidth and reliability. These devices are usually found in embedded control systems, such

Michael Zoumboulakis
Birkbeck College, University of London, WC1N 3QS
e-mail: mz@dcs.bbk.ac.uk

George Roussos
Birkbeck College, University of London, WC1E 7HX
e-mail: gr@dcs.bbk.ac.uk

S. Helmer et al.: Reasoning in Event-Based Distributed Systems, SCI 347, pp. 195–216.
springerlink.com

as Wireless Sensor Actuator Networks (WSAN) [22], with a requirement for timely response to interesting or unusual occurrences in the collected data.

A solution to this problem is to task nodes in the network to push data to base stations. The base stations are connected to powerful desktop-class machines that can perform offline processing. This data harvesting method solves the problem of event detection and pattern recognition but it incurs the significant overhead of expensive radio communication. Multihop links, non-uniform path costs and frequent route failures (cf. [26] for a detailed treatment of these issues) make the global harvesting method impractical. Furthermore, communicating sensor observations using wireless radio is orders of magnitude costlier than local computation [28].

A somewhat better approach involves tasking nodes in the WSAN to perform source-side filtering discarding uninteresting information and only sending packets when interesting observations are recorded. To improve confidence on the event occurrence a node can initiate radio communication to establish whether the event is spatially correlated and consequently produce a notification message for the actuator. The simplest type of filter is realised by thresholds that check whether the sensed observations are above or below a value. Although the simplicity is appealing this technique suffers from severe disadvantages. First, it is sensitive to outliers caused by faulty observations due to inexpensive sensors. Second, it is not capable of handling unknown thresholds such as a case where a user cannot provide a predicate value that distinguishes normal observations from events. Third, it does not scale well as the number of observations and thresholds increase. Last, it is not capable of handling magnitude differences in readings — for instance, two nodes with different temperature sensors will produce readings of different scale requiring two different thresholds.

This chapter discusses methods for event detection in WSANs based on online data mining. First, we depart from the traditional notion of the instantaneous event that can be described by a single occurrence, whether this is a database transaction or a single data point. We introduce the term *pattern* as a finite list of potentially non-unique time-dependent objects. The term sequence as defined by [42] or time series (cf. [14]) refer to the same structure, however the distinguishing factor is that in our case the pattern represents an ordered list of data items that together reveal interesting or unusual activity in the monitored process.

2 Objectives, Motivation and Contributions

Wireless Sensor Actuator Networks (WSANs) have some unique characteristics that distinguish them from other computing devices: first, they are usually powered by commodity batteries or by harvesting energy from the environment. [39], [25]. In both cases, energy is not an abundant resource and its consumption needs to be tightly controlled in a scheduled manner. Related to this restriction is the high energy cost of radio communication which is said to amount to the equivalent of 1000 CPU instructions for sending a bit over the radio [34]. Performing local computations to determine whether a pattern is statistically important or *interesting* is therefore desirable as long as the cost of computation is lower than the cost of transmitting data over the radio. Second, sensor data tends to be noisy due to inexpensive hardware and this brings to surface another requirement: any method for online pattern detection should be tolerant to noise, outliers and missing values while maintaining acceptable *accuracy* and *precision*. Third, WSANs tend to be designed for unattended operation so pattern detection techniques should require little or no human intervention.

In a nutshell, we are examining the problem of efficiently detecting patterns in data that do not conform to a well-defined notion of normal behaviour. The terms *anomaly detection* [7] and *novelty detection* [35] generally encapsulate the objective although the latter extends detection to cover previously unobserved patterns in data. We focus on *online* pattern detection which generally means that detection should take place as close to real time as possible. Since there exist severe resource constraints in a WSAN, pattern detection should investigate segments of sensor data instead of examining data on a global scale which is computationally prohibitive. This is achieved by *windowing*: the application of a sliding window to the streaming sensor data such that a typically smaller data segment is extracted.

The motivating factor for our work is that patterns are ubiquitous across a large number of WSAN applications — a selection of which is reviewed in Section 3 — yet there does not exist a standardised method for their detection. We accept that it is hard to devise an approach that is generalisable across a number of applications since the notion of what constitutes an interesting pattern is subjective and tends to vary according to data characteristics. Despite that, we maintain that there is a significant advantage to be gained by applying pattern detection techniques to WSAN applications. Specifically, we consider methods that borrow concepts from data mining, machine learning and statistics in order to efficiently detect interesting patterns in sensed data. An efficient pattern detection application adds value to a WSAN by contributing to prolonged lifetime of the wireless network and aiding users in identifying which data is important.

Our contribution within the domain of WSAN pattern detection, is a computationally efficient family of methods for pattern detection. These methods cater for pattern detection in both temporal and spatial data, and they require minimal configuration effort. The focus of this chapter is pattern detection in WSANs from a general perspective and in the next section we provide an extensive review of related work that tackles the problem in manners similar to our work. We defer the review of our work [53], [54] to Section 4 where we provide a high-level description of our algorithms. Finally, Section 5 gives our conclusions and identifies open areas of investigation.

Pattern Detection. The term *Pattern Detection* encompasses *Anomaly Detection*, the detection of patterns in data that do not conform to a well-defined notion of normal behaviour, *Novelty Detection*, the detection of previously unforeseen patterns and *Motif Detection*, the detection of patterns in data that recur more often than expected.

3 Review of Work in Pattern Detection in WSAN Data

Before we proceed with the review of related work, some preliminary information related to pattern detection is provided. WSAN data usually has a high degree of spatio-temporal correlation: data from a single node is a linearly ordered sequence where observations are temporally dependent and related in magnitude. Often patterns on the temporal domain differ significantly to their neighbours and this alone may be sufficient for declaring a pattern significantly interesting. On the spatial domain, patterns are usually *collective*: collections of spatially related observations revealing interesting activity in the monitored phenomenon. Sometimes contextual information plays an important role in the task of pattern detection. Consider the example of remote vital signs monitoring of patients at their homes. A pulse oximeter sensor may be used to monitor the heart rate and oxygen saturation level of the patient. A sudden increase in heart rate accompanied by a drop in arterial oxygen saturation may reveal an abnormal situation warrantying concern however if the patient is on an exercise bike or treadmill (context) the pattern may be discarded as a false positive. Another example is data centre monitoring that typically involves monitoring environmental conditions such as temperature and humidity. A sustained increase in temperature that lasts for several minutes may indicate a faulty air conditioning unit but if it occurs within a pre-determined maintenance window it may be flagged as a false positive.

The output of pattern detection is usually a *score* or a *label*. The former can be the output of a distance function such as the Euclidean or the Mahalanobis distance that compares a test pattern to either a reference pattern known to be normal or a user-supplied pattern. Although tasking the user to describe patterns of interest may be desirable in some cases, it leaves the system vulnerable to situations where novel patterns are not detected. Furthermore, it burdens users with description of interesting patterns, a task commonly known in expert systems as the *knowledge acquisition bottleneck* [48]. Conversely, if there is confidence that all normal classes are known a priori then emergent patterns can be checked against the normal classes. In both cases, score or distance based detection involves thresholding to determine whether patterns are normal or abnormal. A simplified scenario is to task nodes to discard patterns with distance below a threshold compared to a collection of reference patterns. Assuming the comparison and distance calculation are cheaper than radio communication, then such a technique promotes network longevity.

Labels may be used in a similar manner to determine whether a pattern is normal, interesting or novel. The number of classes employed is user and application dependent. Clustering approaches may be used to cluster patterns with a degree of membership to each class. Such methods are usually employed for outlier detection [21], [4], [45], [52] a function that is conceptually different to pattern detection. We stress that outlier detection is primarily concerned with detecting single observations that deviate from other observation [18] while pattern detection is concerned with identifying interesting temporally contiguous collections of observations.

The solution space for pattern detection is large with techniques such as hypothesis testing, hidden Markov Models, clustering, density estimation, probabilistic matching, statistical testing, neural networks, Bayesian networks, rule-based systems, expert systems, Nearest Neighbour (NN) based techniques and string matching. However, in the following section we focus on the subset of these techniques with either proven or potential applications for WSANs. The works reviewed are summarised in Table 1 and Section 4 offers a discussion of our work which is based on string matching for the temporal domain and stochastic estimation for the spatial domain.

3.1 Spacecraft and Telemetry Data

We start the discussion with systems aiming to detect interesting patterns in spacecraft observations. In [6] the authors describe how they mine scientific data on-board a spacecraft in order to react to dynamic pattern of interest as well as to provide data summaries and prioritisation. Three algorithms are presented that were used on board the Mars Odyssey spacecraft. The first is designed to detect patterns in images for the purpose of thermal anomaly discovery. A thermal anomaly is defined as a region where the surface temperature is significantly warmer or colder than expected, given its location on the planet, the season, and local topography. The second algorithm was developed to identify polar cap edges and illustrates the importance of online pattern detection: transmitting image data is a costly process for a spacecraft, so it is almost always desirable to prioritise by transmitting only the images that reveal interesting activity, in this instance images containing polar cap edges. This algorithm discovered the water ice annulus south of the north polar cap on Mars. The third algorithm was developed to identify high opacity atmospheric events. The opacity (or optical depth) is a measure of the amount of light removed by scattering or absorption as it passes through the atmosphere. The collection of algorithms employ techniques ranging from trivial and dynamic thresholding to Support Vector Machines (SVMs) and reduced set SVMs. Overall the authors present a very mature approach and they explicitly take into account the processing cost and memory requirements. The single criticism is that the three algorithms seem tailored to the specific problems described — it would be interesting to extend the discussion to potential changes needed to generalise the pattern detection performance of the algorithms.

In [15] the authors describe a system based on Kernel Feature Space and directional distribution, which constructs a behaviour model from the past normal telemetry data and monitors current system state by checking incoming data with the model. This type of system is "knowledge-free" in that is not dependent on a priori expert knowledge. Most modern spacecraft, satellites and orbital transfer vehicles transmit *telemetry* data which is multi-dimensional time series. Usually telemetry data is analysed by ground experts but this paper recognises a recent trend that seeks to apply data mining and machine learning in order to perform online pattern detection. The suggested method works as follows: the multi-dimensional telemetry data is divided into subsets using sliding windows. For each subset the method computes the principal component vector and learns the directional distribution modelled as the von Mises-Fisher (vMF) distribution around the optimal direction computed from the principal component vector. Then it computes the occurrence probability of the principal component vector in relation to the current telemetry data mapped into the feature space. If this probability is below a threshold the data is flagged as anomalous. This system is evaluated against simulator-obtained data for three distinct scenarios involving an orbital transfer vehicle designed to make a rendezvous manoeuvre with the International Space Station. All three scenarios involve faults in the thruster engine and one of the scenarios indicates a scenario where the fault would be hard to determine even by a human expert, as the remaining spacecraft thrusters compensate for the underperforming unit. Although the theory behind their approach is sound it suffers from two disadvantages: first, the evaluation is somewhat limited as it only covers three scenarios with failures of the same component. Second, the computation cost is not explicitly modelled: although the authors mention the assumption of building the model offline using previously collected normal telemetry data, they do not explicitly show the cost of the detection computation.

The approach described in [32] presents three unsupervised pattern detection algorithms that have been evaluated offline using historical data from space shuttle main engine — containing up to 90 sensors — for the objective of future inclusion in the Ares I and Ares V launch systems. The usefulness of an online pattern detection approach is highlighted by the fact that sometimes it takes up to 20 minutes until human experts see data from a spacecraft near Mars, time during which catastrophic events could be prevented if automated pattern detection and actuation was performed on-board. The first algorithm is called Orca and it is based on a nearest-neighbour approach for anomaly detection. The second algorithm (Inductive Monitoring System — IMS) is based on clustering and uses distance to flag a data segment as interesting. The final algorithm uses one-class Support Vector Machines (SVMs): it first maps training data from the original data space into a much higher-dimensional feature space and then finds a linear model in that feature space that allows normal data to be on one side and to be separate from abnormal training data. A limitation of this work is related to the performance of the algorithms which varied across different data sets. As the authors identify, it would be useful if in the future the outputs from the different algorithms were combined to give a more coherent picture on the degree of novelty/anomaly of a pattern.

A somewhat similar system aimed at satellite reliability monitoring is described in [12]. This application aims to automate satellite fault diagnosis, a process currently performed by human experts analysing telemetry data periodically transmitted during a fly-by. The diagnosis of faults from, sometimes limited, sensor data is performed by an expert system. The authors describe how the expert system was built even with limited knowledge and its ability to perform inexact reasoning to accommodate sparse sensors. The disadvantage of the system is inherited from expert systems and it involves the effort necessary in describing all the fault states.

3.2 Environmental Pattern Detection

Moving on to ecological monitoring, the approach described in [2] presents a distributed algorithm for detecting statistical anomalies as well as estimating erroneous or missing data. In short, the proposed method performs automatic inference and prediction based on statistical distributions of differences in measurements between a given node and its neighbours. It is assumed that the observed phenomena are spatiotemporally coherent, so that the measurements at neighbouring nodes show a degree of temporal and spatial correlations. The method works in the following manner: at each timer tick a node calculates the differences between its own measurements and those of its neighbours. It also computes the differences between recent and older (local) measurements. By determining the distribution of the differences it can then perform a *p-test* on each new set of measurements and determine whether it is anomalous by comparing to a threshold. The drawback of this method is that it involves considerable radio communication to spatially compare local with remote readings.

In [37] the authors propose a pattern detection system based on elliptical anomalies which are defined by the ellipsoid or hyperellipsoid caused by the region of distance around the mean (cf. [37] for a detailed definition) of two or more monitored variables. They claim that their system is capable of detecting elliptical anomalies in a distributed manner with exactly the same accuracy as that of a centralised solution. Elliptical anomalies are represented using a hyperellipsoidal model. Given the set of column vectors representing sensor observations, the aim of the approach is to partition the set into two subsets: one containing normal observations and one containing anomalous observations. In simple terms, one way to find such a partition is using (Mahalonobis) distance from the sample mean. Moreover, three categories of elliptical anomalies are defined: first-order, second-order and higher-order elliptical anomalies. The authors suggest that the algorithm for first and second order elliptical anomaly detection can be fully distributed in the network. One criticism of this approach is that computational cost is not explicitly considered although the authors seem to target relatively low-end nodes.

The approach presented in [43] attacks the abstract problem of pattern detection using a density test for distributional changes. The main idea is that new, potentially multidimensional, data can be tested against a baseline. For this given baseline data set and a set of newly acquired observations, a statistical test is performed that aims to determine if data points in the newly acquired set were sampled from the same underlying distribution that produced the baseline set. The test statistic is distribution-free and largely based on kernel density estimation and Gaussian kernels. The baseline distribution is inferred using a combination of this kernel density estimator with an Expectation Maximisation algorithm. The strength of the approach is that it seems capable of detecting patterns occurring at multiple data dimensions simultaneously. However it suffers from the disadvantage of high resource and computational requirements making it potentially prohibitive for low-end sensor nodes.

An approach that takes a machine-learning viewpoint to the pattern detection problem is described in [27]. More specifically, the authors describe an

Instance-Based Learning (IBL) model that aims to classify new sets of observations according to their relation to a previously acquired reference data. By storing historical examples of normal observations, the normalcy of emerging observations can be assessed. The authors recognise that such an approach inherently suffers from the high cost overhead of storing multiple "normal" instances and addresses the problem with a combination of instance selection and clustering algorithms that reduce the dictionary size of normal data. The approach also includes a noise suppression filter that removes noise while performing feature selection from the data. The output of the algorithm is a binary decision determining the input as normal or abnormal. Although the authors claim that this solution is highly generalisable to multiple domains, it remains to be determined whether it is suitable for severely resource-constrained environments where the storage size is extremely limited and the cost associated with reading and writing to it is relatively high. Furthermore, there are cases where the user requires more than binary classification and this approach does not cater for approximate detection.

The approach described in [1] presents a model-based predictive system that aims to detect and predict patterns as river flood events in developing countries by deploying sensor networks around the basin area of rivers. The simplest model is based on statistical methods such as linear regression using a portion of data known to be normal. The project aims to cover vast geographical regions of approximately 10,000 km^2 and predict a pattern of interest using a distributed model driven by the collected data. The main drawback of this approach is that it assumes a tiered architecture where resource-constrained sensor nodes transmit summaries and statistics of raw data to a set of computation nodes. The latter determine the correctness of the data, feeds it to the model for prediction and may request additional data from sensors to reduce uncertainty. A somewhat similar tiered system is PRESTO [10] which employs ARIMA (Auto Regressive Integrated Moving Average) time series forecasting models and performs anomaly detection by comparing predicted values to sensor observations.

The work described in [51] introduces contour maps to display the distribution of attribute values in the network and employs contour map matching to determine whether a user-supplied pattern matches node-produced observations. The application scenario is event detection in coal mines, monitoring for the occurrence of gas, dust and water leakage as well as high/low oxygen density regions. A limitation of this approach is that it assumes users capable of perfectly describing the pattern of interest as distributions of an attribute over space and variations of this distribution over time incurred by the event.

3.3 Spatial Structure Pattern Detection

A method aimed at pattern detection over trajectory data is described in [5] where the authors propose a distance metric to determine similarity between trajectory subsequences. Trajectory data is usually a sequence of longitude and latitude readings obtained from Global Positioning System (GPS) readings. Detecting patterns in trajectory data has attracted considerable interest recently due to its numerous applications ranging from remote monitoring of elderly patients to military detection of enemy movements. The contribution of the method is an algorithm that builds local clusters for trajectories in a streaming manner. In detail, it focuses on the problem of determining if a given time window has fewer than *k* neighbours in its left and right sliding windows, and if so it flags it as an interesting pattern. To improve on efficiency, the authors introduce a data structure (Vantage-Point Tree) that facilitates piecewise rescheduling of cluster joins. Overall, this is a promising approach and the experimental results show the efficiency of the method requiring less than 0.5 milliseconds of processing time for newly acquired observations. The disadvantage of the approach is that it has only been evaluated with offline data lacking the robustness gained by field experiments on real WSAN nodes.

The approach described in [41] focuses on detecting interesting patterns across linear paths of data. A linear path refers to a path represented by a line with a single dimensional spatial coordinate marking an observation point, such as mile markers on motorways or locations along gas pipe lines. Potential applications of pattern detection on this domain would be the discovery of unusual traffic patterns such as accident hubs or proactive infrastructure monitoring on gas or water pipes, tunnels, bridges and so on. The proposed approach is called Scan Statistics for Linear Intersecting Paths (SSLIP) and is in fact a family of algorithms that employ statistical methods for online data mining. With respect to detection of interesting patterns, the method relies on the calculation of an *unusualness* metric that indicates the likelihood ratio or degree of unusualness of a given window of observations in comparison with past data. The windows with highest likelihood ratios are flagged as interesting or unusual. The weakness of this approach is that it is employing heavyweight, in terms of processing, techniques such as Monte Carlo simulations to determine the significance of patterns. Furthermore, the authors clearly state that this method should be used in conjunction with domain experts for the identification of interesting patterns — this restricts the autonomous detection that is highly desired in WSANs and inevitably makes the process somewhat more interactive.

Another relatively interactive approach is described in [47] that introduces *rare category detection*. The main contribution of the proposed method is the capability of detecting both statistically significant and interesting patterns. The central premise of category detection involves tasking a user to label patterns with predetermined categories. A pattern that doesn't belong to any category is novel and is classified to a new category. The presented algorithm aims to identify the rare categories in data using hierarchical mean shift, a variation of the mean shift algorithm which is an iterative procedure that shifts each data point to the average of data points in its neighbourhood [8]. The hierarchical variation involves the iterative application of mean shift with increasing bandwidths, such that a hierarchy of clusters at different scales can be created whilst annotating each cluster with an anomaly score. The two main weaknesses of this approach is that it can be computationally expensive and that it requires a small degree of human interaction. Although the user need not provide information about the data such as the number of categories or their prior probabilities, he or she is still required to classify a statistically significant pattern as interesting.

3.4 Data Centre Monitoring and Context-Aware Computing

Moving on to the emerging research area of data centre monitoring and design, the research described in [36] proposes an online temporal data mining solution to optimise the performance of data centre chillers. Sensor nodes deployed in data centres produce time series data that describe environmental conditions that usually vary in time according to the load of individual servers, storage and networking equipment. This type of system exemplifies the automated control that is typical of a WSAN: according to when/what interesting patterns are sensed a decision must be made to turn on/off chillers, select a utilisation range and generally react to cooling demands. The authors focus their efforts on *motif mining*, that is the identification of frequently occurring temporal patterns. First, they obtain a symbolic representation of the time series data using the Symbolic Aggregate Approximation (SAX) [30] algorithm that is also employed by our own method (Section 4). A Run-Length Encoding (RLE) of the symbol sequence is performed in order to note when transitions from one symbol to another occur. Generally speaking, RLE is a technique that reduces the the size of a repeating string of characters by replacing one long string of consecutive characters with a single data value and count [3]. Frequent episode mining is conducted over the sequence of transitions to identify the underlying efficiency profile of the data centre under different environmental circumstances. This allows the formation of dynamic models of data centre chillers and formulation of control strategies.

A somewhat similar approach [33] targets online novelty detection on temporal sequences utilising Support Vector Regression (SVR) to model the time series data. The authors introduce detection with a confidence score indicating the confidence of the algorithm for the degree of novelty. There is particular consideration to the event duration which is rarely known in advance and must be selected with some care to avoid missing events or spurious matches. The significance of the training window is also identified and its relationship to resource requirements is one the weaknesses of the approach. Another weakness is the limited evaluation that does not provide concrete evidence of the efficacy of SVR for novelty detection.

The approach of [17] largely targets context-aware computing and specifically mining for anomalous structures in observations representing human interactions with their environment. The authors propose the use of a *Suffix Tree* data structure to encode the structural information of activities and their event statistics at different temporal resolutions. The method aims to identify interesting patterns that either consist of core structural differences to previously normal behaviour or differences based on the frequency of occurrence of motifs. With regards to anomalous pattern detection, the authors adopt the view that given a set of normal activity sequences A, any subsequence of events is classified as normal as long as it occurred in A. Naturally, this relies on training data used to construct the dictionary of legitimate behaviour. Sequences are classed as anomalous using a match statistic computed over the suffix tree. The strength of the approach is that suffix tree traversal can be conducted in linear time, however dynamic suffix tree construction and update is not always suitable to platforms with severe resource constraints that lack dynamic memory allocation capabilities.

3.5 Network Monitoring and Intrusion Detection

Moving on to approaches that aim to ensure WSAN stability, the work described in [11] introduces Artificial Immune Systems (AIS) for misbehaviour detection. AIS are inspired from principles that the human immune system uses for the detection of harmful agents such as viruses and infections. Since WSANs typically lack the infrastructure readily available in wired networks, it is somewhat easier for attackers to maliciously modify a WSAN either by dropping packets or compromising the routing topology. In this context, an interesting pattern would be a sequence of unusual actions for one or more nodes. The method aims to facilitate local learning and detection on WSAN nodes using a gene-based approach. Each node maintains a local set of detectors that is produced by negative selection from a larger set of randomly generated detectors tested on a set of self strings. These detectors are then used to test new strings that represent local network behaviour and detect non-self strings. The approach has been evaluated using MAC protocol messages and has shown encouraging results. Although such an approach is valuable for the detection of local patterns that indicate misbehaviour over some layer of the OSI (Open Systems Interconnection) reference model stack, it is not clear how it can be generalised to apply to sensor-acquired observations.

A somewhat related approach [31], targets intrusion detection in WSANs from samples of routing traffic. First, feature selection of traffic and non-traffic related data is performed in order to learn the distribution of values affecting routing conditions and traffic flows. Second, anomaly detection is performed locally by taking a window of data and examining it against previously collected normal data. This window contains samples that are mapped to points in a feature space and subsequently analysed together with their surrounding region in the feature space. If a point lies in a sparse region of space is classified as anomalous, using a fixed-width clustering algorithm. The method has the capability of detecting previously unseen (novel) patterns while respecting resource constraints by performing detection locally but at the same time shows a weakness in detecting slow attacks happening gradually over a long time scale. Finally, evaluation is somewhat limited and performed exclusively via simulations that do not comprehensively model a large number of attacks.

The approach described in [19], proposes a Principal Component Analysis (PCA) method for detecting anomalies with complex thresholds in a distributed manner. Nodes send their readings to a coordinator node that is responsible for firing a trigger based on the aggregate behaviour of a subset of nodes. The individual nodes perform filtering such that they send readings only when measurements deviate significantly from the last transmitted data. With respect to detection, they propose two window

Table 1 Comparison of pattern detection techniques for sensor data

Approach	Basis	Application	Strength	Weakness
[6]	Dynamic Thresholding and SVMs	Spacecraft image data	Respects constraints	Algorithm tailor-made to problem
[15]	Kernel functions	Spacecraft telemetry data	No prior knowledge required	Comp. costs not explicitly modelled
[32]	NN/Clustering /SVMs	Spacecraft engine data	Pattern mining in data from up to 90 sensors	Performance varies across data sets
[12]	Expert System	Satellite telemetry data	Inexact reasoning	Knowledge acquisition bottleneck common in Expert Systems
[37]	Elliptical Anomalies	Abstract/Environmental data	Distributed solution with same accuracy as centralised	Comp. cost not explicitly modelled
[2]	Statistical p-value tests	Ecological anomaly detection	Automatic Inference & Prediction	Radio communication cost
[43]	Kernel density estimators	Abstract/Multidimensional data	Detects patterns occurring at multiple dimensions simultaneously	High computational requirements
[27]	ML/Instance-based classification	Abstract/Temporal sequences	Noise suppression filter	Binary classification/No approximate detection
[5]	NN/Clustering	Trajectory pattern detection	Efficient processing time for classifying new observations	No evaluation of online operation
[41]	Scan Statistics for Linear Intersecting Paths	Unusual traffic pattern discovery	Unusualness metric	High computational requirements
[47]	Hierarchical Mean Shift	Abstract/Rare category detection	Distinguishes statistically significant and interesting patterns	High computational requirements
[36]	Symbolic conversion/Run-length encoding	Data centre monitoring	Dynamic modelling of data centre chillers	Bias towards motifs rather than novelties
[33]	Support Vector Regression	Abstract/Temporal sequences	Confidence score	High computational costs related to large training windows
[17]	Subsequence matching using Suffix Trees	Context-aware computing	Linear time matching	Not suitable for dynamic tree updates
[11]	Artificial Immune Systems	Misbehaviour detection on network data	Efficiency of local detectors	Not generalisable to sensor-acquired data
[31]	Fixed-width clustering	Network Intrusion Detection	Detects previously unseen patterns	Misses slow-occurring patterns
[19]	Principal Component Analysis	Distributed network pattern detection	Source-side filtering	Relies on coordinator node (SPOF)
[1]	Model-based/Statistical regression	River flood pattern events	Distributed data-driven model	Assumes presence of a resource-rich tier
[51]	Contour-map matching	Coal-mine monitoring	SQL extensions allows users to specify events as pattern	Relies on prior-distribution knowledge by user

triggers that are persistent threshold violations over a fixed or varying window of time series data. Although the approach is aimed at detecting unusual network traffic patterns, it could apply to certain WSAN applications. The main criticism is that it creates single points of failure by assigning the coordinator role to nodes.

4 Symbolic and Stochastic Pattern Detection

Having covered sufficient background and related work, we now focus on our approach to pattern detection for extremely resource-constrained devices. Before we proceed with the details of our approach, we enumerate the requirements for a pattern detection solution in resource-constrained Wireless Sensor Actuator Networks:

- It must be capable of detecting patterns both on the temporal and spatial domain, since sensornets usually monitor phenomena on a spatio-temporal scale.
- It must be capable of detecting both previously unseen patterns and user-submitted patterns with exact and approximate matching semantics.
- It must tolerate outliers, noise, missing values as well as scale differences in the sensor-acquired data.
- It must explicitly take into account the resource-constraints of the execution environment. Specifically, the solution targets low-end nodes such as the TMote Sky [40] and the battery-free Intel WISP [38].
- It must fit well with the existing communication paradigms without requiring any modifications to lower layer protocols.
- It must scale as the number of user-submitted patterns increase as well as the network size grows.

Over the following sections we provide an initial discussion of how the above requirements are addressed. For experimental evidence and in-depth coverage the interested reader is referred to our previous work [55], [53], [56], [54].

4.1 Temporal Pattern Detection Using a Symbolic Representation

First, we convert the pattern detection problem to a pattern matching problem. A mature symbolic representation algorithm — *SAX* [30] — used for numerous data mining tasks is employed to convert sequences of numeric sensor observations to character strings. SAX is a linear time algorithm that first coverts a time series to an intermediate representation using Piecewise Aggregate Approximation (PAA). The resulting string is obtained by converting the PAA representation using a table lookup. Due to space restrictions we will not discuss SAX here in further detail, but we refer to interested reader to the published literature on SAX (cf. [30, 24, 23]. The reasons we convert numeric data to symbols are threefold:

1. A symbolic representation opens up access to a wide variety of mature string matching algorithms,
2. A symbolic representation achieves data reduction and thus requires less space achieving radio communication and storage savings, and

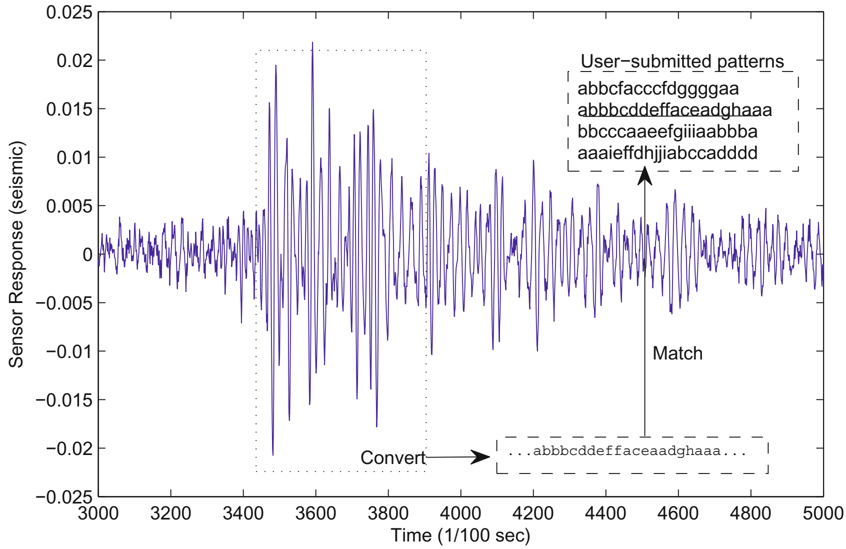

Fig. 1 Example of converting a time series segment to symbols and matching against a database of user-submitted patterns. The user typically enters a pattern as a numeric sequence. This sequence is converted to a string either by an application that acts as the interface between the WSAN and the user, or by the WSAN node itself.

3. A string distance function is defined that lower-bounds the Euclidean distance allowing to perform matching entirely on the symbolic representation without loss of information.

The symbolic conversion algorithm is capable of handling a degree of missing values, noisy data and outliers in the numeric input data. However, no inference of missing values is performed and a high number of missing values would inadvertently affect the matching performance of the algorithm. With regards to detection we offer three methods:

1. *Approximate* or *Exact* Pattern Detection where the user knows in advance the pattern of interest and wishes to be informed when the sensor-acquired data converts to patterns that match the reference pattern either approximately or exactly (example in Figure 1)
2. *Non-parametric* Pattern Detection, where the user need not supply any information in advance but instead the algorithm trains on a window of normal sensor data by constantly observing distances in temporally adjacent strings corresponding to temporally neighbouring time series sequences
3. *Probabilistic* Pattern Detection where a Markov Model is built by monitoring individual character transitions. Strings with improbable character transitions are labelled as unusual.

Further to the above, we cater for Dynamic Sampling Frequency Management (DSFM). Similar to Non-parametric Pattern Detection, this algorithm involves a a training phase to learn the sensed process dynamics and use it to make autonomous local decisions to dynamically increase or decrease the frequency in which the observations are acquired from the sensors. Dynamically adjusting the frequency enables network nodes to conserve energy in periods of relative inactivity for times when interesting patterns occur. We recognise that certain applications can have specific sampling frequency requirements depending on the periodicity of the signal. Having these requirements specified as a sampling interval rather than an absolute value, can help nodes select an appropriate frequency via the DSFM capability, relaxing the pre-deployment need for complex signal processing.

The scalability requirement is met by introducing a *Suffix Array* structure for storage and fast searching of user-submitted patterns. In terms of fitting with standard methods, our approach uses a standard Publish/Subscribe interface that employs state-of-the-art WSAN communication protocols without requiring any modifications or incurring any communication overhead. The computational efficiency of the algorithm is achieved by aggressive optimisations (cf. [56] for a detailed discussion) using integer arithmetic and verified by measuring the execution time on nodes from operational WSAN systems. The additional cost of converting numeric sequences to strings is almost negligible: typically 11 milliseconds of CPU time are required for the conversion of a numeric sequence for a window of 40 data points to a string and this time includes a string distance comparison to a user-submitted pattern. Naturally, larger windows are possible, for instance to capture long-duration pattern events, and can be easily introduced either at pre-deployment or injected dynamically at runtime.

In the case of user-submitted patterns, one point of interest is the relation between the length of the user pattern and the window size employed by the symbolic conversion algorithm. In this context, the length of the user pattern relates to the length of the resulting string, in other words the output of the symbolic conversion. The user submits a pattern in a numeric form and this converts to a string either by the WSAN node itself or by an application that acts as an interface between the WSAN and the user. There exist two possibilities: in Approximate Pattern Detection the length of the user-submitted pattern must be equal to the length of the string produced by a WSAN node. This is due to the string distance function employed for comparison that only accepts strings of equal length. However this can be adapted by using alternative distance functions that can accept strings of different length — the Sequence Alignment algorithms (cf. [16]) are such examples of comparison functions that accept strings of different length. The second option relates to Exact Pattern Detection where the Suffix Array structure is employed. In this case the window size is affected in the following manner: if a user submits a pattern smaller than the resulting string produced by the WSAN node, matching is unaffected since the Suffix Array stores all the suffixes of the user submitted pattern. Conversely, if a user submits a pattern larger than the resulting string produced by the WSAN node, exact matching is affected since strings of different length can never match exactly. To resolve this, the WSAN node adjusts the window size such that the resulting string length is equal to the largest user-submitted string.

Further to the performance evaluation on operational WSAN nodes, in order to refine the granularity of measurements we have opted to simulate the operation of a WSAN by replaying data sets collected in-situ and emulating the operations of a node in software. This approach was used with three data sets representing distinct case studies: first, normalised seismometer and acoustic data from a volcano monitoring application ([50]) was used to quantitatively evaluate the performance against a large number of organic events of varying durations. Indicative results are shown in Table 2 where the double compression refers to running two instances of the symbolic conversion algorithm with different compression settings for accuracy. In fact, the lightweight nature of the approach allows users to execute it simultaneously with different configuration values without penalising performance. This is a significant benefit in cases where the pattern event duration or characteristics are unknown. Second, a case study using environmental data from an indoor network [20] was used to evaluate the performance against data imperfections such as outliers and missing values. We found the accuracy of the algorithm unaffected by imperfections and at the same time real and synthetic patterns were detected on temperature, humidity and light data. Third, ECG and accelerometer data from the UCR data mining archive [46] was used to evaluate the applicability of the algorithm to potential pervasive healthcare and context-aware deployments. Again, we found the algorithm responded well in detecting unseen patterns and pinpointing the change from normal to anomalous. Due to space limitations we cannot discuss the experimental result in great detail however we invite the interested reader to review our results ([53]).

Finally, we have recently adapted the algorithm for the battery-free Intel WISP platform that harvests electromagnetic energy from RFID readers. We have modified the original WISP design by adding a supercapacitor to store harvested energy so that a node can continue operating when it moves outside the range of an RFID reader. We intend to use the pattern detection algorithm for the purpose of activity recognition and more specifically teaching younger children about movements and elementary mechanics (cf. [55] for more details). This verifies the versatility of the pattern detection algorithms for different tasks and their suitability for the extremely resource-constrained platforms.

Table 2 Summary of quantitative detection accuracy results with one and two compression settings.

Compression Setting	**Detection Accuracy**	
Single compression (4 : 1)	Detected: 733 out of 947	**77.4%**
Double compression (2 : 1 and 4 : 1)	Detected: 878 out of 947	**92.7%**

4.2 Spatial Pattern Detection and Source Location Estimation

For the detection of patterns on the spatial domain, we focus on a specific class of events namely dispersion of pollutants from a single static point source. We assume a WSAN is deployed in the area of the dispersion plume where nodes individually detect the presence of the pollutant in the atmosphere using the algorithm of the previous section or some other method. Since this pattern has a strong spatial element, local detection is not sufficient in itself. The aim is to initiate a "walk" of the network such that a coarse estimate of the source's location is iteratively computed.

The algorithm is similar to local gossip approaches (cf. [29] for the *Trickle* algorithm, an example of a gossip protocol for WSANs) and works by instantiating a Kalman filter that iteratively predicts the state of the dispersion process at neighbouring nodes. At the beginning of the process, the originating node makes an estimate of the observations at its one-hop neighbours (line 2, Algorithm 1). Since no other information is available this estimate is a linear transformation of the local reading. The neighbour that minimises the error is selected as the next hop and receives the necessary Kalman filter parameters and values to continue the process (line 8, Algorithm 1). A geometric computation is employed to take into account the neighbourhood consensus before making the routing decision in order to reduce message cost at each hop (not shown in Algorithm 1 for simplicity). This works in the following manner: after a small (i.e. ≤ 10) number of hops, a convex hull is evaluated for the coordinates of the nodes that participate in the estimation. Provided that the estimation process begins at nearby locations, the convex hull can be computed cheaply without a significant communication overhead. The mean direction of movement is given by calculating the centroid of the convex hull. At this stage only a coarse quadrant direction is necessary: the Cartesian coordinates quadrant in which the majority of nodes participating in the estimation process believe the source is located. This geometric computation adds robustness to the algorithm such that erroneous local routing decisions can be overridden by the majority consensus. Experimental evaluation has shown estimation accuracy of up to 97.33%, where an estimate is considered accurate if it is within 6 meters from the actual position of the source.

The iterative estimation process stops when the estimation error becomes unacceptably high (lines 13-15, Algorithm 1) which indicates that either the node has approached a region close to the source or that it has entered a region where observations differ greatly from the estimate of the process. The process halts and the coordinates and intensity of the last node in the path become the final estimate.

This spatial detection not only detects the pattern but provides useful metadata with respect to the pattern location and intensity. Such estimation tasks are common in applications such as [9] concerned with the *Inverse* problem: given some sensor observations, the goal is to estimate the source location. We have evaluated the spatial detection algorithm over grid and random distributions of different densities and we have found it outperforms a maximum selection algorithm (that selects neighbours with the higher reading) while it is competitive with other heavy-weight

Algorithm 1. Spatial Pattern Detection (SED) Location Estimation Algorithm

1: **variables** Estimate Error Covariance P, Measurement Noise Variance R, Process Variance Q, State Transition Matrix A, Measurement Matrix H, Initial Estimate $\hat{x}_k{}^-$, maxhopcount=1, netpath[], counter $c = 0$;
2: Project state estimate $\hat{x}_k{}^-$ ahead (cf. Equation 4.9 [49])
3: Project error covariance $P_k{}^-$ ahead (cf. Equation 4.10 [49])
4: Task *unvisited* neighbours within maxhopcount to report measurement.
5: **for** (each of replies received) **do**
6: calculate innovations $(z_k{}^{(i)} - H\hat{x}_k{}^-)$
7: **end for**
8: Select as next hop the node that minimises the innovation.
9: Compute the Kalman gain (cf. Equation 4.11 [49])
10: Correct (Update) estimate with measurement $z_k{}^{(i)}$ (cf. Equation 4.12 [49])
11: Correct (Update) the error covariance P_k (cf. Equation 4.13 [49])
12: Compute relative error.
13: **if** abs(relative error) $>=$multiple$\cdot(\mathbb{E}[Rel\ Error])$ **then**
14: **exit**
15: **else**
16: Add local address to netpath[c] and increment c.
17: Send algorithm parameters to selected node (line 8) and task it to start at Line 1.
18: **end if**

location estimation approaches. For further information and experimental results the interested reader is referred to [54].

5 Conclusions

In this chapter we have outlined the need for efficient pattern detection in Wireless Sensor Actuator Networks (WSANs) that lack core infrastructure services such as reliable communication and time synchronisation. The majority of middleware approaches that attempt to deal with the pattern detection problem from a composite event calculus perspective are not suitable for severely resource-constrained execution environment. Instead, pattern detection techniques that adapt statistical, machine-learning and data mining approaches are much more suitable.

We presented a large collection of pattern detection approaches from different application domains that address the problem using a multitude of techniques. A universal solution to the problem capable of detecting patterns in different types of data is extremely difficult since application requirements vary. To partially address this problem, we described in Section 4 a data-mining inspired technique that employs string matching and is capable of detecting patterns in sensor data of different modalities across both temporal and spatial domains. This method requires little or no configuration and therefore fits well the long-term vision that anticipates WSANs comprised of millions of inexpensive nodes. Furthermore, it leverages the development of existing state-of-the-art communication methods using standard interfaces such as Publish/Subscribe for the notification of interesting patterns and without requiring any modifications to lower layer protocols.

The discussion of Pattern Detection for WSANs leaves a few directions open for further exploration. First, the impact of local coordination in relation to Non-Parametric and Probabilistic Pattern Detection has to be investigated further. Local coordination refers to geographical adjacent WSAN nodes exchanging information that facilitates the training phase of the algorithms. Second, we aim to investigate a direction based on real-time classification of WSAN data. There is preliminary work in this area utilising string algorithms [44] and numerous applications exist ranging from assisted diagnosis [13] to augmenting learning processes for children [55]. The final research direction involves the investigation of mixed methods where multiple detection algorithms run in parallel within a WSAN in order to improve detection accuracy. Addressing the above directions through evaluation on operational WSANs will extend the work on Pattern Detection and benefit users of reactive applications across a number of application domains.

References

1. Basha, E.A., Ravela, S., Rus, D.: Model-based monitoring for early warning flood detection. In: SenSys 2008: Proceedings of the 6th ACM conference on Embedded network sensor systems, pp. 295–308. ACM, New York (2008)
2. 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: Aspnes, J., Scheideler, C., Arora, A., Madden, S. (eds.) DCOSS 2007. LNCS, vol. 4549, pp. 223–239. Springer, Heidelberg (2007)
3. Bose, R.: Information theory, coding and cryptography. Tata McGraw-Hill, New York (2002)
4. Branch, J., Szymanski, B., Giannella, C., Wolff, R., Kargupta, H.: In-network outlier detection in wireless sensor networks. In: 26th IEEE International Conference on Distributed Computing Systems, ICDCS 2006, pp. 51–59 (2006)
5. Bu, Y., Chen, L., Fu, A.W.-C., Liu, D.: Efficient anomaly monitoring over moving object trajectory streams. In: KDD 2009: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 159–168 (2009)
6. Castano, R., Wagstaff, K.L., Chien, S., Stough, T.M., Tang, B.: On-board analysis of uncalibrated data for a spacecraft at mars. In: KDD 2007: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 922–930 (2007)
7. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. ACM Computing Surveys (2009)
8. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 17(8), 790–799 (1995)
9. Chin, J.-C., Yau, D.K.Y., Rao, N.S.V., Yang, Y., Ma, C.Y.T., Shankar, M.: Accurate localization of low-level radioactive source under noise and measurement errors. In: SenSys 2008: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, pp. 183–196. ACM, New York (2008)
10. Desnoyers, P., Ganesan, D., Li, H., Li, M., Shenoy, P.: PRESTO: A predictive storage architecture for sensor networks. In: Tenth Workshop on Hot Topics in Operating Systems, HotOS X (2005)

11. Drozda, M., Schaust, S., Szczerbicka, H.: Is AIS based misbehavior detection suitable for wireless sensor networks. In: Proc. IEEE Wireless Communications and Networking Conference (WCNC), Citeseer (2007)
12. Durkin, J., Tallo, D., Petrik, E.J.: FIDEX: An expert system for satellite diagnostics. In: In its Space Communications Technology Conference: Onboard Processing and Switching, pp. 143–152 (1991) (see N92-14202 05-32)
13. Dutta, R., Dutta, R.: Maximum Probability Rule based classification of MRSA infections in hospital environment: Using electronic nose. Sensors and Actuators B: Chemical 120(1), 156–165 (2006)
14. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. SIGMOD Rec. 23(2), 419–429 (1994)
15. Fujimaki, R., Yairi, T., Machida, K.: An approach to spacecraft anomaly detection problem using kernel feature space. In: KDD 2005: Proceedings of the eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, pp. 401–410 (2005)
16. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge (1997)
17. Hamid, R., Maddi, S., Bobick, A., Essa, I.: Unsupervised analysis of activity sequences using event-motifs. In: VSSN 2006: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks, pp. 71–78 (2006)
18. Hawkins, D.M.: Identification of outliers. Monographs on applied probability and statistics. Chapman and Hall, Boca Raton (1980)
19. Huang, L., Garofalakis, M., Hellerstein, J., Joseph, A., Taft, N.: Toward sophisticated detection with distributed triggers. In: MineNet 2006: Proceedings of the 2006 SIGCOMM workshop on Mining network data, pp. 311–316. ACM, New York (2006)
20. Intel. Lab Data, Berkeley (2004), `http://db.csail.mit.edu/labdata/labdata.html`
21. Janakiram, D., Reddy, V.A., Kumar, A.: Outlier detection in wireless sensor networks using bayesian belief networks. In: First International Conference on Communication System Software and Middleware, Comsware 2006, pp. 1–6 (2006)
22. Karpiński, M., Cahill, V.: Stream-based macro-programming of wireless sensor, actuator network applications with SOSNA. In: DMSN 2008: Proceedings of the 5th workshop on Data management for sensor networks, pp. 49–55. ACM, New York (2008)
23. Keogh, E., Lin, J., Fu, A.: HOT SAX: Efficiently Finding the Most Unusual Time Series Subsequence. In: IEEE International Conference on Data Mining, pp. 226–233 (2005)
24. Keogh, E., Lonardi, S., Ratanamahatana, C.A.: Towards parameter-free data mining. In: KDD 2004: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 206–215. ACM, New York (2004)
25. Kompis, C., Aliwell, S.: Energy Harvesting Technologies to Enable Wireless and Remote Sensing — Sensors & Instrumentation KTN Action Group Report (June 2008), `http://server.quid5.net/ koumpis/pubs/pdf/energyharvesting08.pdf`
26. Krishnamachari, B.: Networking Wireless Sensors. Cambridge University Press, Cambridge (2005)
27. Lane, T., Brodley, C.E.: Temporal sequence learning and data reduction for anomaly detection. ACM Trans. Inf. Syst. Secur. 2(3), 295–331 (1999)
28. Levis, P., Culler, D.: Maté: A Tiny Virtual Machine for Sensor Networks. In: ASPLOS-X: Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems, New York, NY, USA, pp. 85–95 (2002)

29. Levis, P., Patel, N., Culler, D., Shenker, S.: Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In: NSDI 2004: Proceedings of the 1st Conference on Symposium on Networked Systems Design and Implementation, vol. 2, USENIX Association, Berkeley (2004)
30. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: DMKD 2003: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, pp. 2–11. ACM, New York (2003)
31. Loo, C.E., Ng, M.Y., Leckie, C., Palaniswami, M.: Intrusion detection for routing attacks in sensor networks. International Journal of Distributed Sensor Networks 2(4), 313–332 (2006)
32. Oza, N., Schwabacher, M., Matthews, B.: Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring. Journal of Aerospace Computing, Information, and Communication 6(7), 464–482 (2007)
33. Ma, J., Perkins, S.: Online novelty detection on temporal sequences. In: KDD 2003: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 613–618. ACM, New York (2003)
34. Madden, S., Franklin, M.J., Hellerstein, J.M., Hong, W.: The design of an acquisitional query processor for sensor networks. In: SIGMOD 2003: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 491–502. ACM, New York (2003)
35. Markou, M., Singh, S.: Novelty detection: a review–part 1: statistical approaches. Signal Processing 83(12), 2481–2497 (2003)
36. Patnaik, D., Marwah, M., Sharma, R., Ramakrishnan, N.: Sustainable operation and management of data center chillers using temporal data mining. In: KDD 2009: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1305–1314 (2009)
37. Rajasegarar, S., Bezdek, J.C., Leckie, C., Palaniswami, M.: Elliptical anomalies in wireless sensor networks. ACM Trans. Sen. Netw. 6(1), 1–28 (2009)
38. Intel Research. WISP: Wireless Identification and Sensing Platform (2008), `http://seattle.intel-research.net/wisp/`
39. Roundy, S., Wright, P.-K., Rabaey, J.: Energy Scavenging for Wireless Sensor Networks: with Special Focus on Vibrations, 1st edn. Springer, Heidelberg (2003)
40. MoteIV (later renamed to Sentilla). TMote Sky Datasheets and Downloads (2008), `http://www.sentilla.com/pdf/eol/tmote-sky-datasheet.pdf`
41. Shi, L., Janeja, V.P.: Anomalous window discovery through scan statistics for linear intersecting paths (SSLIP). In: KDD 2009: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 767–776 (2009)
42. Sipser, M.: Introduction to the Theory of Computation. PWS Pub Co, Boston (1996)
43. Song, X., Wu, M., Jermaine, C., Ranka, S.: Statistical change detection for multi-dimensional data. In: KDD 2007: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 667–676. ACM, New York (2007)
44. Stiefmeier, T., Roggen, D., Tröster, G.: Gestures are strings: efficient online gesture spotting and classification using string matching. In: BodyNets 2007: Proceedings of the ICST 2nd international conference on Body area networks, pp. 1–8 (2007)
45. Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online Outlier Detection in Sensor Data Using Non-Parametric Models. In: Dayal, U., Whang, K.-Y., Lomet, D.B., Alonso, G., Lohman, G.M., Kersten, M.L., Cha, S.K., Kim, Y.-K. (eds.) VLDB, pp. 187–198. ACM, New York (2006)

46. Riverside University of California. The UCR Time Series Data Mining Archive (2008), http://www.cs.ucr.edu/~eamonn/TSDMA
47. Vatturi, P., Wong, W.-K.: Category detection using hierarchical mean shift. In: KDD 2009: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 847–856 (2009)
48. Wagner, W.P.: Issues in knowledge acquisition. In: SIGBDP 1990: Proceedings of the 1990 ACM SIGBDP conference on Trends and directions in expert systems, pp. 247–261 (1990)
49. Welch, G., Bishop, G.: An Introduction to the Kalman Filter. Technical Report 95-041. Chapel Hill, NC, USA (1995)
50. Werner-Allen, G., Dawson-Haggerty, S., Welsh, M.: Lance: optimizing high-resolution signal collection in wireless sensor networks. In: SenSys 2008: Proceedings of the 6th ACM conference on Embedded network sensor systems, New York, NY, USA, pp. 169–182 (2008)
51. Xue, W., Luo, Q., Chen, L., Liu, Y.: Contour map matching for event detection in sensor networks. In: SIGMOD 2006: Proceedings of the 2006 ACM SIGMOD international conference on Management of data, pp. 145–156. ACM, New York (2006)
52. Zhang, J., Wang, H.: Detecting outlying subspaces for high-dimensional data: the new task, algorithms, and performance. Knowledge and Information Systems 10(3), 333–355 (2006)
53. Zoumboulakis, M., Roussos, G.: Efficient pattern detection in extremely resource-constrained devices. In: SECON 2009: Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 10–18 (2009)
54. Zoumboulakis, M., Roussos, G.: Estimation of Pollutant-Emitting Point-Sources Using Resource-Constrained Sensor Networks. In: Trigoni, N., Markham, A., Nawaz, S. (eds.) GSN 2009. LNCS, vol. 5659, pp. 21–30. Springer, Heidelberg (2009)
55. Zoumboulakis, M., Roussos, G.: In-network Pattern Detection on Intel WISPs (Demo Abstract). In: Proceedings of Wireless Sensing Showcase (2009)
56. Zoumboulakis, M., Roussos, G.: Integer-Based Optimisations for Resource-Constrained Sensor Platforms. In: Hailes, S., Sicari, S., Roussos, G. (eds.) S-CUBE 2009. LNICIST, vol. 24, pp. 144–157. Springer, Heidelberg (2010)