Anomaly Detection in Wireless Sensor Networks in a Non-Stationary Environment

Colin O'Reilly, *Member, IEEE,* Alexander Gluhak, Muhammad Ali Imran, *Senior Member, IEEE,* and Sutharshan Rajasegarar, *Member, IEEE*

***Abstract*—Anomaly detection in a wireless sensor network (WSN) is an important aspect of data analysis in order to identify data items that significantly differ from normal data. A characteristic of the data generated by a WSN is that the data distribution may alter over the lifetime of the network due to the changing nature of the phenomenon being observed. Anomaly detection techniques must be able to adapt to a non-stationary data distribution in order to perform optimally. In this survey, we provide a comprehensive overview of approaches to anomaly detection in a WSN and their operation in a non-stationary environment.**

***Index Terms*—Wireless sensor networks, anomaly detection, outlier detection, non-stationary, concept drift, distributed computing**

I. Introduction

LARGE scale monitoring applications such as smart city realisations [1], environmental monitoring [2], [3], industrial monitoring [4], internal building monitoring [5], [6] and surveillance [7], [8] provide valuable information for intelligent decision making and smart living. However, collecting data from such applications can pose a significant challenge due to the size and location of the monitored area, the environmental conditions and the deployment timescale. WSNs provide a platform for solving this monitoring challenge, which are low cost, easy to deploy, and require little or no maintenance during the lifetime of the network.

A WSN is formed using interconnected nodes that automatically configure themselves. There are three important elements that characterize a WSN node, namely one or more sensors, a processing unit and a transceiver. Sensors in the node allow the measurement of parameters of the physical surroundings. A microprocessor allows intelligent computation to be performed on the node, and a wireless radio receiver enables communication among the neighbouring nodes. Wireless communication between neighbouring nodes allows the automatic formation of a network without the need for a costly wired infrastructure. The sensor nodes in a WSN are resource constrained, including limited processing and storage, limited energy resource, short communication range and low bandwidth [9].

A key function of a WSN is the analysis of data that is generated in the form of measurements by sensor nodes. One objective of data analysis is anomaly detection. The aim of anomaly detection is to identify data that do not conform to the patterns exhibited by the majority of the data set [10]. An anomaly or outlier (these terms are used interchangeably in this paper) is defined as "an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data" [11]. Algorithms that perform anomaly detection construct a model using a set of data measurements. The model is then used to classify data as either *normal* or *anomaly*. Measurements collected by sensors form a time-ordered sequence of data. During the lifetime of data collection, the underlying phenomenon that is being measured may alter. This will cause a change in the distribution of the data and thus the data distribution will no longer be a *stationary data distribution* but will be a *non-stationary data distribution*. If a system has a stationary data distribution then no temporal correlation exists as all data are equally related and drawn independently and identically distributed (i.i.d) from a stationary distribution. In this case, the model of the data from which to identify anomalies only needs to be constructed once. For optimal performance, the model should be constructed after enough data are available in order to have a good generalization error on the testing data set. In an environment with a non-stationary data distribution, it is necessary to construct a new model at certain intervals in order to account for changes in the data distribution. An assumption is made that the data are temporally correlated, with correlation increasing as temporal distance decreases. Therefore, in order to achieve the best generalization error, the training set needs to be formed from data that are temporally close to the data that will form the testing set.

Previous surveys on anomaly detection techniques have focused on data sets where the underlying data distribution is assumed to be stationary. These surveys have detailed the usage of statistical or machine learning techniques that are used to identify anomalies. Chandola *et al.* [10] survey the application domains in which anomaly detection is applied, and the statistical and machine learning techniques that are used to detect anomalies. Anomaly detection in the specific application domain of WSNs has been surveyed by Rajasegarar *et al.* [12], [13] and Zhang *et al.* [14] where the focus is on anomaly detection techniques that operate within the resource

Manuscript received August 13, 2012; revised March 7, 2013 and September 12, 2013. We acknowledge the support from the REDUCE project grant (No. EP/I000232/1) under the Digital Economy Programme run by Research Councils UK – A cross-council initiative led by EPSRC.

C. O'Reilly, A. Gluhak and M. A. Imran are with the Centre for Communication Systems Research, University of Surrey, Guildford, United Kingdom (e-mail: {c.oreilly, a.gluhak, m.imran}@surrey.ac.uk).

S. Rajasegarar is with the Department of Electrical and Electronic Engineering, University of Melbourne, Melbourne, Australia (e-mail: sraja@unimelb.edu.au).

Digital Object Identifier 10.1109/SURV.2013.112813.00168

constraints and distributed architecture of a WSN. Anomaly detection in WSNs from the perspective of security is surveyed by Xie *et al.* [15].

This survey takes a different view of current state-of-the-art anomaly detection techniques in WSNs. We examine anomaly detection from the perspective of operation within a non-stationary environment. In many application domains normal behaviour can evolve, therefore current normal behaviour might not be sufficiently representative of future normal behaviour [10]. We believe that this is the first survey that analyses the operation of anomaly detection algorithms in a non-stationary environment. State-of-the-art algorithms are surveyed and techniques are examined which are able to identify changes in the data distribution and update a model to include new information and remove old information. Our survey aims to highlight current approaches to the problem, point out areas which are lacking, and recommend areas for future research.

The remainder of this paper is organized as follows. Section II gives the fundamental characteristics of anomaly detection in a WSN. Section III discusses non-stationary distributions of data and its effect on anomaly detection. Section IV presents a taxonomy for the classification of techniques and a workflow of their operation. Section IV also presents methods to evaluate anomaly detection techniques. Section V discusses change detection. Section VI and VII survey the update to a model. Section VIII provides a discussion on the shortcomings of current research and recommends areas for future research. Section IX concludes this paper.

II. Fundamentals of Anomaly Detection in Wireless Sensor Networks

This section presents the fundamental characteristics of WSNs and anomaly detection in WSNs. The environment in which a WSN is deployed is discussed. In addition, the characteristics of data and definitions of anomalies in WSN data are provided.

A. The Environment of the WSN

WSNs may be deployed in harsh, unattended environments for significant periods of time where it is impossible to carry out maintenance on the nodes after installation. Therefore it is important that algorithms deployed on sensor nodes are self-managing and can adapt to changing environments.

The constrained environment of a WSN impacts on anomaly detection algorithms. Node constraints on computational power and memory mean that algorithms for anomaly detection should have low computational complexity and occupy little memory space. In addition, there are constrained energy resources on the wireless sensor node and this impacts on communication between nodes. The use of the wireless radio receiver consumes significantly more energy than any other component on the sensor node. Pottie and Kaiser [16] state that the cost of transmitting 1Kb a distance of 100 metres has the same energy cost as performing 3 million instructions on a general-purpose processor. In general, the cost of receiving is comparable to the cost of transmitting. Thus there is a requirement to minimize the number and length of transmissions in order to conserve energy.

B. Correlation of Data

In order to ensure satisfactory coverage of a monitored area, a spatially dense deployment is required [17], [18]. This deployment leads to multiple nodes sensing the same event. This causes data on the nodes to have the same underlying data distribution and thus spatial correlation exists.

In addition to spatial correlation, temporal correlation of data can occur. Temporal correlation arises when there exists a predictable relationship between sequential data points. Data measurements on an individual node can be temporally correlated due to the nature of the phenomenon that is being observed; for example, temperature measurements may exhibit a predictable rising and falling pattern each day.

Finally, spatial-temporal correlation of data can occur in WSNs where data collected on different nodes and at different times exhibit a predictable relationship. In a densely deployed WSN there will be correlation of data if a set of nodes within spatial proximity are measuring the same phenomenon.

The spatial, temporal and spatial-temporal correlation of data can be exploited to identify an anomaly and determine its cause. Anomalies caused by errors occur independently, whereas anomalies caused by events exhibit spatial and/or temporal correlation. Vuran *et al.* [19] study these correlations in order to utilize them to reduce energy consumption in a WSN. The spatial correlation of data can lead to a distributed learning structure where information describing the data one node is experiencing can be communicated to other nodes for them to incorporate into their models of data. The spatial correlation of data ensures that experiences of one node might be similar to that of other nodes and hence it is useful for nodes to share identified characteristics of the data. The temporal correlation of data can require that more recent data are used to construct models to classify current data as the correlation between data measurements can decrease as a function of time.

C. Distributed Learning in a WSN

In WSNs, data are gathered at nodes which are dispersed in a physical environment but connected through the medium of a wireless interface and a routing protocol. Individual sensor nodes measure local environmental conditions, and therefore data are dispersed across the network. The aim of anomaly detection is to identify the outliers in the data sets on the individual nodes. If the assumption is made that there is spatial correlation of data then information exchanged between nodes may increase the accuracy in the detection of anomalies. Learning in a distributed environment can be divided into three distinct categories; *local*, *distributed* and *central*.

In the local learning approach, the model of the local data is learned and only anomalies based on this local model can be detected. This method avoids costly transmission of data measurements or model data between nodes and thus can be a more energy efficient anomaly detection method. However, each node obtains an independent classifier and the spatial correlation of data is not used. Events learned about in another part of the network can not be recognized by the node unless the node has also encountered them.

The centralized approach communicates all data to a gateway node. A model is constructed using all the data, and anomalies are identified in the entire data set. This has the advantage that the gateway node can be more powerful and can therefore use more computationally complex anomaly detection algorithms on the data. However, the communication cost in transmitting a local node's data measurements to a central node can be prohibitive. In addition, there are scalability issues as the ratio of nodes to gateways increases. Finally, there has been an increase in the timeframe for detection for online algorithms due to the delay introduced by the transmission of data to a central node.

Distributed learning attempts to limit the transmissions between nodes while building a model constructed from information from a number of nodes within the network. Nodes run local instances of an anomaly detection algorithm in order to infer patterns from data measurements arriving at the node from the sensors. Nodes then exchange information about local models in order to build a global model that encompasses data from other nodes in the network. Summarized information, which can take the form of model parameters and/or anomalies, is transmitted as opposed to data. Therefore there is a reduction in transmissions. However, an event on a node can still influence model construction on another node. Often there is a trade-off between local anomaly detection and communication with other nodes in the network. The more communication that occurs, the more the performance of the algorithm tends to that of a centralized model with improved global detection accuracy. The distributed model can vary in its aim. Some algorithms aim to infer from data in a neighbourhood of sensors and others aim to construct a local model that tends towards the model that would have been constructed by a centralized approach. Algorithms can distribute information in a number of ways. There can be a simple exchange of the mean value of a data set or more resource intensive operations such as broadcasting anomalies in order to determine how other models classify them. This can lead to vast differences in the amount of transmissions that occur, typically the most expensive energy operation for a sensor node.

D. Anomaly Detection in WSNs

It is possible to view anomaly detection from two different aspects which drives the manner in which anomalies are identified. The first aspect is data fault detection which seeks to identify data points that have been generated in error. The second aspect is novelty detection which seeks to identify data instances that are indicative of a (possibly rare) event of interest that needs to be analyzed further. Thus anomaly detection can be aimed at identifying data faults, identifying novel instances, or at identifying both and distinguishing between the two.

Data faults are measurements that are inconsistent with the nature of the phenomenon being observed [22]. Identifying this type of error is important as they can cause data to be added to the data set that does not correspond to the underlying distribution. Anomalies influence the quality of the data that are provided to anomaly detection algorithms. Data that includes anomalies can introduce skew or additional complexity into the model. This causes difficulty in constructing a model for the data. In addition, in a distributed environment where data instances are transmitted between nodes, removing data faults can save energy that otherwise would have been wasted in their transmission.

Sharma *et al.* [22] studied sensor faults in WSNs and showed that there was a large variation in the number of faults occurring in real-world WSN implementations. Faults ranged from less than 0.01% to 15–35%. Spatial and temporal correlation among the faults was discovered in only one implementation and this was due to the batteries in spatially correlated nodes dying at approximately the same time.

Novel behaviour in a system is also a source of anomalies. This can be seen as an event that is rare or not in the normal range of activity and has not been incorporated into the distribution of the data set. Yoon *et al.* [23] used anomalies in data from a system monitoring pipelines in an oilfield to identify novel behaviour such as pipe blockage and leakage. Further actions may be taken on data instances identified as novel.

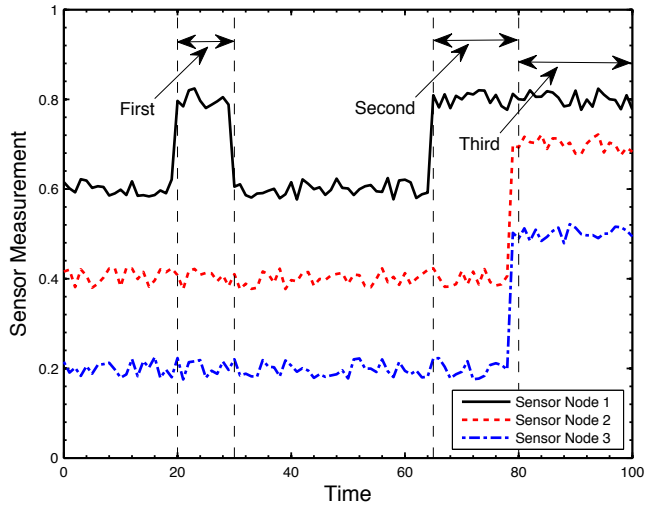
Anomalies can occur due to different causes. The phenomena that is being monitored may have an unusual element that causes the data generated to significantly differ from normal behaviour. In addition, a security threat can cause anomalies to be generated. Intrusion detection is a technique which monitors the behaviour of a system. The aim is to differentiate normal from anomalous behaviour in order to identify security threats such as a denial of service attack or a node compromise [24]–[27].

Anomalies with different causes may have different characteristics. However, it is useful to categorize anomalies based on various properties. Properties include how much it differs from normal data instances, the number of occurrences, and the location of anomalies within the WSN. Rajasegarar *et al.* [12], [20] emphasize the distributed nature of a WSN and define anomalies based on their correlation with data on other sensor nodes.

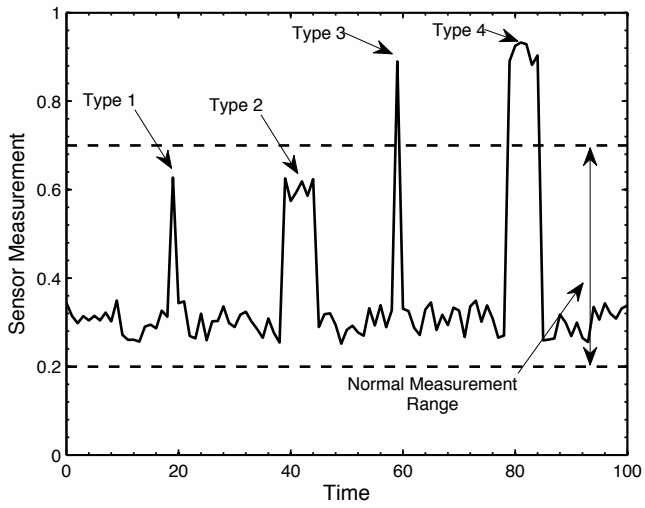
- First Order Anomalies: Partial data measurements are anomalous at a sensor node
- Second Order Anomalies: All data measurements at a sensor node are anomalous
- Third Order Anomalies: Data from a set of sensor nodes are anomalous

Fig. 1a displays first, second and third order anomalies. We consider 3 nodes in a larger sensor network. It is assumed that the measurements at nodes 1, 2 and 3 should be in the region of 0.6, 0.4 and 0.2 respectively. This occurs in the first 20 time periods. Sensor node 1 displays a first order anomaly from time period 20 to 30 when there are a number of anomalous readings, whereas the other two sensor nodes display normal data. From time period 65 to 100, the data measurements from sensor node 1 are anomalous. These are second order anomalies where all data at the node is anomalous. From time period 80 to 100 sensor nodes 1, 2 and 3 all have third order anomalies, where the data from this set of nodes are all classified as anomalies compared to the normal data from other sensor nodes in the WSN.

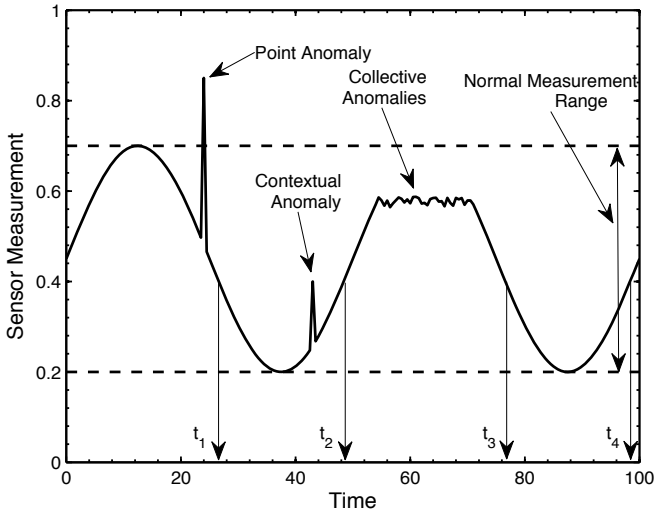
Zhang *et al.* [21] classify anomalies based on the cause of the anomaly on a local node.



(a) First, Second and Third Order Anomalies [12], [20]



(b) Type 1, 2, 3 and 4 Anomalies [21]



(c) Point, Contextual and Collective Anomalies [10]

Fig. 1. Different Definitions of Anomalies in WSN data sets.

- Type 1: Incidental absolute errors: A short-term extremely high anomalous measurement
- Type 2: Clustered absolute errors: A continuous sequence of *type 1* errors
- Type 3: Random errors: Short-term observations not lying within the normal threshold of observations
- Type 4: Long term errors: A continuous sequence of *type 3* errors

Fig. 1b displays the anomalies defined by Zhang *et al.*. At time period 20 a type 1 anomaly occurs as this data instance differs significantly from the normal data, but lies within the observation range. From time period 40 to 45 an extended burst of type 1 anomalies occurs. These are termed type 2 anomalies. At time period 60, a measurement occurs that significantly differs from the normal data and is outside the observation range. This is termed a type 3 anomaly. Finally, from time period 80 to 85 an extended burst of type 3 anomalies occurs. These are termed type 4 anomalies.

In the area of security in WSNs three anomaly types, namely point, contextual or collective anomalies [10], are used to compare techniques [15].

- Point anomaly: An individual data instance that is considered anomalous with respect to the data set.
- Contextual anomaly: A data instance that is considered an anomaly in the current context. In a different context the same data instance might be considered normal.
- Collective anomalies: A collection of related anomalies.

Fig. 1c displays the anomalies defined by Chandola *et al.* [10]. A point anomaly occurs at time period 24 where the data instance is anomalous with respect to the entire data set. At time period 43 a contextual anomaly occurs which is anomalous at this time, but would not be considered anomalous had it occurred at time t_1, t_2, t_3 or t_4. Finally, collective anomalies occur in the time period 54 – 71. Collective anomalies are a set of data instances that exhibit a pattern, however, they are anomalous with regard to the entire data set.

The three definitions of anomalies in WSNs provide an insight into the characteristics of anomalies that can occur in data sets. However, these definitions are not comprehensive for all anomalies that may occur in WSNs.

III. Non-Stationary Data Distribution

In this section the terms stationary and non-stationary are defined. Assumptions made in machine learning and pattern recognition techniques are examined and the effect that a non-stationary data distribution has on these assumptions is discussed. In addition, the effect that a non-stationary data distribution has on anomaly detection in WSNs is explored. The section concludes with examples of non-stationary data sets from real-world WSN deployments.

A. Machine Learning and Non-Stationary Data Sets

A fundamental assumption of standard machine learning and pattern recognition theories is that the data used in a training set are drawn from a stationary data distribution, and the testing set will also be drawn from the same distribution [28]. Thus it is assumed that $P_{train}(x) = P_{test}(x)$ [29]. This is often unrealistic in real-world environments [28]. A change in the data distribution can cause a model trained with data from a previous distribution to become suboptimal for the current distribution. Application domains such as network monitoring, economic and financial data analysis generate data that are changing in its distribution as time progresses [30]. Changes can occur for several reasons, including changes in the fundamental natural process which generates the observation.

Anomaly detection in data sets has been widely examined in the machine learning community. The main focus of attention in WSNs has been on stationary data sets where the data distribution is assumed to be constant over time. Algorithms either ignore non-stationary distributions or assume that a periodic retraining will account for change, for example [20], [31]–[33]. Due to the assumption that a training data set and a testing data set are drawn from a stationary data distribution, if the data distribution alters between the drawing of the training and testing data set, the model will not be correct for the testing data set. This will lead to a degradation in performance of the anomaly detector. O'Reilly *et al.* [34] studied a data set in which the anomaly rate varied and showed that if the model does not adapt to the varying rate, performance degrades.

Two approaches to the problem of anomaly detection in a non-stationary environment can be defined. One method is to monitor the data distribution, if a change is detected the model is retrained. Another approach is to make an assumption that the training and test inputs have different probability distributions, but that the conditional distribution of the output values given the input values is not altered. This is known as covariate shift adaptation [35]. In this survey we focus on the former technique, the identification of change in the data distribution and effective and efficient adaptation to the change.

B. Stationary and Non-Stationary Processes

Alterations in the underlying phenomenon that is being observed can cause changes to the data that are being generated by the sensor nodes in a WSN. Kelly *et al.* [36] identified three ways in which a non-stationary distribution may exhibit change through the use of Bayes Theorem.

Bayes Theorem and the posterior probability states that for a data instance $\mathbf{x}$ and class ω

$$P(\omega|\mathbf{x}) = \frac{P(\mathbf{x}|\omega)P(\omega)}{P(\mathbf{x})} \tag{1}$$

Firstly, the class priors, $P(\omega)$, may change overtime. Secondly, the distributions of the classes might change, where $P(\mathbf{x}|\omega)$ alters over time. Finally, the posterior distributions of the class may change, $P(\omega|\mathbf{x})$ [36].

Not all changes will cause the classifier to be incorrect for the current data distribution. If the class priors, $P(\omega)$, and the likelihood of observing a data point within a particular class, $P(\mathbf{x}|\omega)$, alters, the posterior distribution of class membership, $P(\omega|\mathbf{x})$ might not change. This is termed *virtual drift* [37].

Other changes will alter the performance of the classifier that was trained using a data set from a different distribution. *Concept drift* [38] is defined as changes in the posterior distribution of the class (concept) membership as time progresses where $P_{t+1}(w|\mathbf{x}) \neq P_t(w|\mathbf{x})$ [39]. Furthermore, concept drift is defined as a gradual change to the target variable, and *concept shift* is defined as a more abrupt change to the target variable [38], [40], [41].

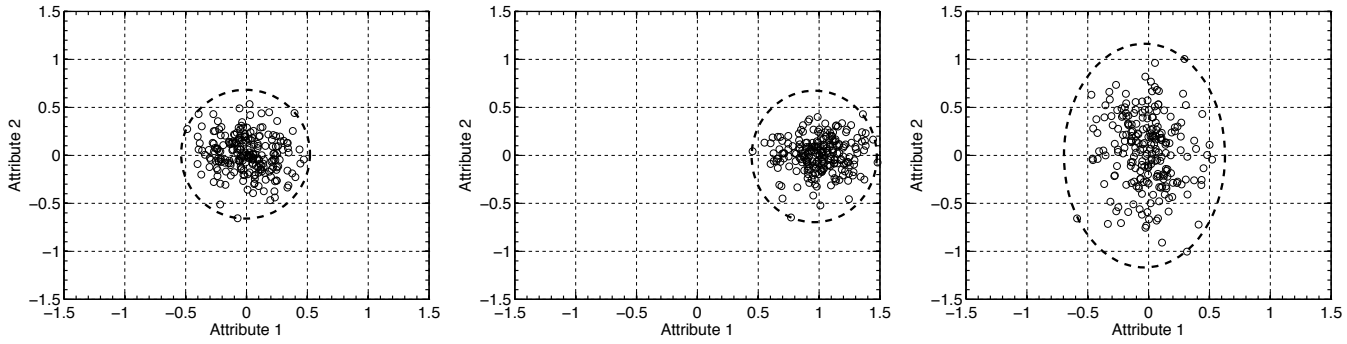
It has been proposed that it is not necessary to differentiate between changes to the concept and changes to the data distribution, as both alterations require a model to be updated [41]. Therefore, we examine methods by which an adaptation can be made by an anomaly detector, regardless of the nature of the change to the data. We use the more general term non-stationary (distribution) rather than referring to specific types of change to the concept or data distribution. It is necessary that effective anomaly detection algorithms are able to adapt to non-stationary data distributions in order to construct accurate models which minimize the error on unseen data [42]–[44].

C. Anomaly Detection in a Non-Stationary Environment

Previously, the nature of data in a non-stationary environment was discussed. Attention is now turned to the application domain of anomaly detection and the effect of a non-stationary distribution.

Anomaly detection differs from supervised two-class classification problems. Anomaly detection uses one-class classification where one concept class, rather than two, is defined. The purpose is to classify a data vector as either belonging to the class, a *normal* data vector, or not belonging to the class, an *anomaly* data vector. If we define the concept as the target variable that the algorithm is trying to model, then anomaly detection aims to model the concept in order to identify data that does not belong to it. Therefore, for the normal class N and the anomaly class A, $P(\omega) = P(N)$ and $P(A) = 1 - P(N)$. The posterior probability of the normal class membership is $P(N|x)$, which defines the class boundary for the normal data.

A non-stationary distribution can affect anomaly detection in two ways:



(a) $X, Y \sim \mathcal{N}(0, 0.04)$ (b) $X \sim \mathcal{N}(1, 0.04)\, Y \sim \mathcal{N}(0, 0.04)$ (c) $X \sim \mathcal{N}(0, 0.04)\, Y \sim \mathcal{N}(0, 0.16)$

Fig. 2. The effect of a non-stationary distribution on the class boundary. Changing the data distribution will result in changes to the boundary location and shape.

- Change in the distribution of the normal class which affects the class boundary of the normal data – an alteration in $P(N|\mathbf{x})$
- Change in the ratio of anomalies to normal data – an alteration to the prior $P(A)$ (and consequently $P(N)$)

1) Effect on the Normal Class Boundary: In a WSN the data set is formed of sensor measurements of a phenomenon. Changes in the phenomenon will cause changes in the data distribution which will result in a shift in the boundary of the normal class.

Defining this mathematically, the training and testing set will consist of a time-ordered sequence of data vectors $X = \{\mathbf{x_i} : i = 1, 2, 3, ..., n\}$ each of which is p variate data vector $\mathbf{x_i} = (x_{i1}, x_{i2}, x_{i3}, ..., x_{ip})$, $i = 1, ..., n$. The probability that a data vector belongs to the normal class is stated as $P(N|\mathbf{x}) = P(\mathbf{x}|N)P(N)/P(\mathbf{x})$. If the distribution is non-stationary, there will be an alteration in the posterior distribution of the normal class $P_{t+1}(N|\mathbf{x}) \neq P_t(N|\mathbf{x})$.

Fig. 2 shows the effect of a changing data distribution on the class boundary. If we consider the initial data distribution to be that of Fig. 2(a), the class boundary of the normal data is centred at the origin. In Fig. 2(b) we observe that the mean of the distribution for attribute 1 has shifted from 0 to 1 performing a transformation of the class boundary along the x-axis. Another example of a change that can occur is in Fig. 2(c) where there has been a change in the standard deviation of the distribution of attribute 2 causing a vertical expansion of the class boundary.

An alteration in the class boundary of the normal data can cause problems for anomaly detection algorithms. A model built using a training set generated from a previous distribution may no longer be optimal for the current distribution causing it to misclassify normal data as anomalies and vice versa.

2) Effect on the Anomaly Rate: If the data distribution is non-stationary, the rate that anomalies occur in the data set can be affected. Some algorithms use the anomaly rate as a threshold in order to determine the class boundary for the normal data.

The class prior probabilities are defined as $P(\omega)$. In the application domain of anomaly detection, there is only one class, this is the class of normal data. Therefore the class prior $P(N)$ also determines the anomaly rate as $P(A) = 1 - P(N)$. A change in $P(N)$ will cause a change in the anomaly rate $P(A)$. This is an important consideration in anomaly detection as certain algorithms make an assumption that the anomaly rate is known and is specified as a parameter during model construction. If the anomaly rate varies, anomalies can be misclassified as normal data and vice versa.

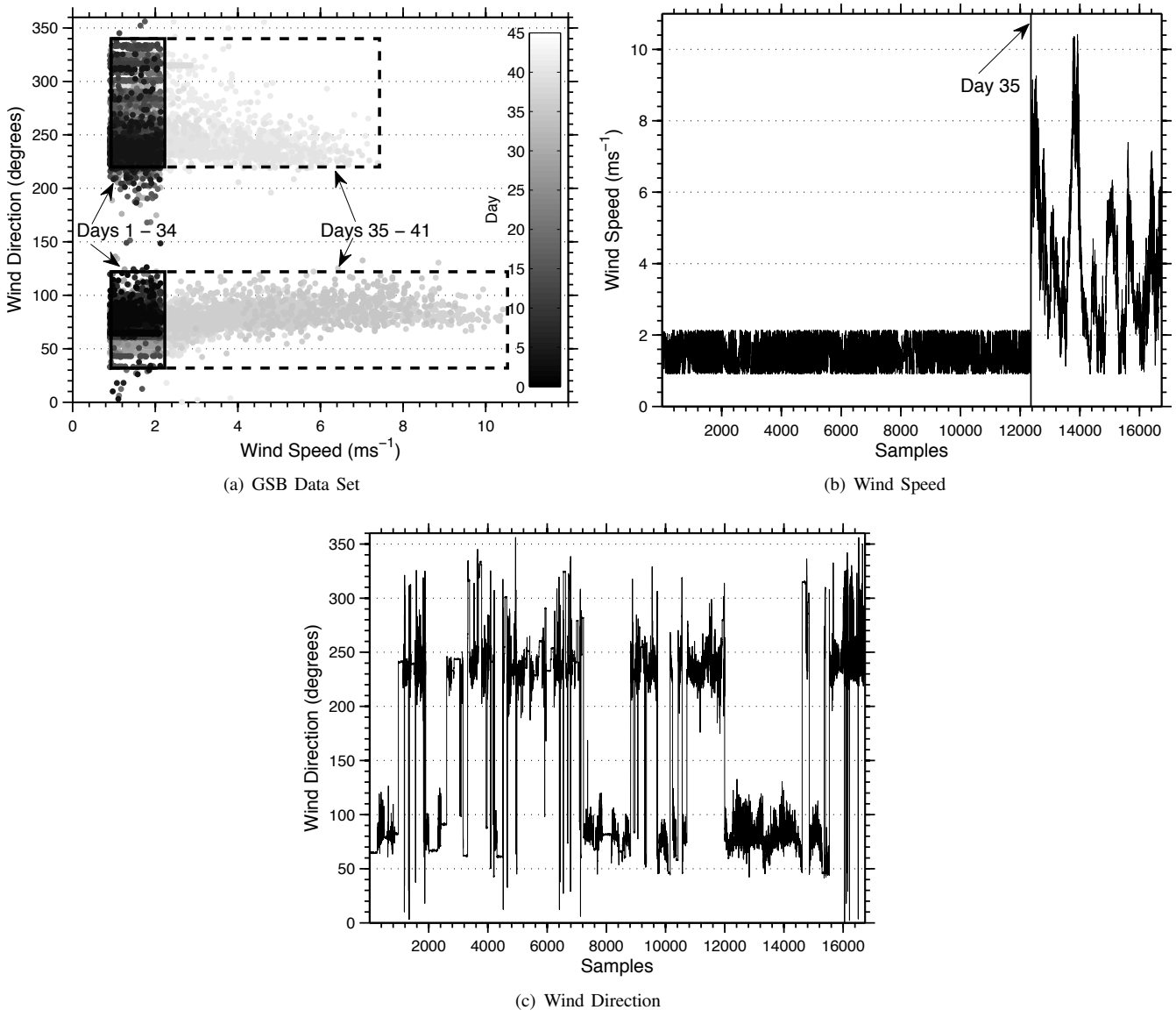
3) Effect on the Anomaly Class Boundary: In anomaly detection, the class boundary of the anomalous data is not usually taken into consideration. The one-class classification approach assumes that anomalies are under sampled and it is not possible to extract information about the anomaly data distribution from the available anomalous data instances [45]. Therefore, no attempt is made to model the anomaly class. Due to this, changes in the class boundary of the anomalies, $P(A|\mathbf{x})$, will not affect classification performance.

D. Examples of Non-stationary Data in Real-World Data Sets

We have shown that if data are non-stationary in nature, a change in the data distribution will occur. In this section we provide details of several real-world data sets that are non-stationary.

One example of such a data set is the Grand-St-Bernard (GSB) data set. The data was gathered from a set of 23 sensors deployed in the Grand-St-Bernard pass between Switzerland and Italy in 2007 [46]. Two sensor measurements, wind data in the form of speed in ms^{-1} and the angle of the wind direction in degrees, are shown to exhibit a non-stationary data distribution. There is an abrupt change, a concept shift, over the measured period causing a change in the normal data class boundary. An examination of the wind measurements for node 4, Fig. 3(a), shows that the data distribution over the first 34 days is stationary and occupies two well-defined areas. However, from day 35 there is a sustained increase in the wind speed occurring in the same direction as previously. Examining the two sensor data streams separately, Fig. 3(b) and 3(c), the wind speed is in the range 1 and 2 ms^{-1} for the first 120,000 samples, which is until day 34. From sample 120,000, there is an increase in the wind speed over the remaining 4000 samples, with the wind speed increasing to a maximum of 10 ms^{-1}. The wind direction follows a similar pattern over the entire period. Other nodes from the deployment in GSB show similar characteristics for wind data.

Another data set that shows non-stationarity is the Intel Berkeley Research Laboratory (IBRL) data set. This is used



(a) GSB Data Set

(b) Wind Speed

(c) Wind Direction

Fig. 3. A real-world non-stationary data set. Wind speed and wind direction data from node 4 of the GSB data set.

by Zhang *et al.* [47] with an adaptive anomaly detector where updates to the model are required in order to account for changes in the data distribution. The IBRL data set is also used by Moshtaghi *et al.* [48] in the study of updates to an iterative elliptical boundary tracking algorithm.

Non-stationary data sets taken from sensor data include a signalled road intersection [49] where sensors provided data on traffic volume and which are used to predict the volume of traffic in the next hour. In addition, sensor measurements from weather data have been used to study incremental learning in non-stationary environments. Elwell *et al.* [39] studied incremental learning in a non-stationary environment on sensor data from a weather station at the Offutt Air Base in Bellevue, Nebraska [50].

IV. Analyzing Anomaly Detection Techniques Designed for Non-Stationary Environments

In this section a taxonomy is presented in order to classify anomaly detection techniques that are surveyed in Sections V, VI and VII. In addition, a work flow is presented that details how the different components operate together in order to provide anomaly detection in a non-stationary environment. Finally, the issue of performance evaluation is addressed. Methods of measuring the performance and complexity of an anomaly detector are detailed.

A. Taxonomy for Anomaly Detection in a Non-Stationary Environment

Previous work on taxonomies for anomaly detection techniques have focused on the statistical or machine learning technique that is used to identify anomalies in data sets. Chandola *et al.* [10] categorize the methods into the main machine learning categories such as classification-based and nearest neighbour-based approaches. This taxonomy is continued in the work of Rajasegarar *et al.* [12], [13], Zhang *et al.* [14] and Xie *et al.* [15] whose surveys focus on anomaly detection in WSNs.

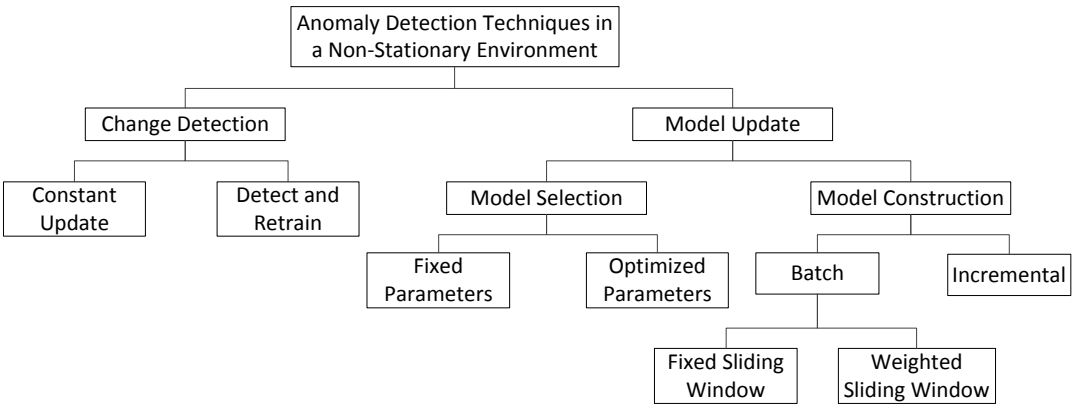


Fig. 4. Taxonomy of anomaly detection techniques in a non-stationary environment.

As illustrated in Fig. 4, the taxonomy categorizes methods that anomaly detection techniques can use in order to adapt to a non-stationary distribution. There are two steps that need to be performed; the first is *change detection* where the aim is to identify changes in the data distribution in order to determine when a model update is required. The change detection techniques can be categorized as either *constant update* or *detect and retrain* [51].

The second step is *model update*, where the model is reconstructed in order to adapt to changes in the data distribution. Two techniques are part of this process, *model selection* and *model construction*. *Model selection* is the process by which a model is chosen for a specific training set that is by some measure optimal. The techniques can be classified as *fixed parameters* and *optimized parameters*. *Model construction* describes how a model is constructed using a training set that differs from the previous model. This can be further divided into two categories *batch* and *incremental*. Batch is categorized as either *fixed sliding window* or *weighted sliding window*.

State-of-the-art approaches to anomaly detection will implement *change detection*, *model update – model selection* and *model update – model construction*. Our survey is structured in this manner. Techniques within these categories are mutually exclusive, however, the categories themselves are not. We select anomaly detection methods that highlight how a particular technique is being performed, and note it may perform other techniques in other categories. Table I in Section VIII summarizes the algorithms and the techniques that they implement.

B. Workflow for Anomaly Detection in a Non-Stationary Environment

The constituent modules that form a system that is able to perform anomaly detection in a non-stationary environment are now studied. Fig. 5 provides a visual representation of the process.

1) Change Detection: Sensors will measure a phenomenon that is generating data with a non-stationary distribution. Multiple sensors on a node will form X_t, an n dimensional data vector at time t, where n is the number of sensors on a node.

If constant update is performed, no monitoring is performed on the data distribution. A model update will be scheduled at regular intervals.

If detect and retrain is performed, the data will be monitored in order to determine whether there has been a change in the data distribution. When the change detection algorithm determines that a significant change has occurred in the data distribution, an update to the model will occur.

2) Training Set Formation: The training set for the model is formed from the data vectors $\{X_1, X_2, X_3, ...\}$. If a sliding window is used, it will frame the data vectors that are to be used for model construction. For the next model, the sliding window will shift n data vectors allowing the n oldest data vectors to be removed from the window, while n new data vectors will be added to the window.

3) Model Selection: Next the parameters are determined for the training set. If the parameters are fixed they will have been determined previously at deployment and only one model can be constructed for the current training set. If the parameters are to be optimized, the algorithm will determine the optimal parameters for the current training set.

4) Model Construction: Model construction occurs next. The parameters determined previously are used in the construction of the model. If a batch update occurs, the previous model is discarded and the model for the new training data is constructed. For an incremental update, the previous model, $model(t-1)$ and the n new data vectors are used to construct a new model, $model(t)$.

5) Anomaly Detection: The new model, $model(t)$, is used as the anomaly detector for data. The data vectors $X_{t+1}, X_{t+2}, X_{t+3}, ...$ generated from the process form the testing data set and are labelled as either *normal* or *anomaly*. An assumption is made that the testing data set is drawn from the same data distribution as the training data set.

C. Evaluation of Anomaly Detection Techniques

In order to evaluate anomaly detection techniques, several metrics are defined. An anomaly detection algorithm will classify a data vector formed of sensor measurements as either normal or anomaly. Comparing the assigned labels to the ground-truth labels, a false positive is defined as a normal data vector incorrectly labelled as an anomaly, a true positive is defined as an anomaly correctly identified. From this, two

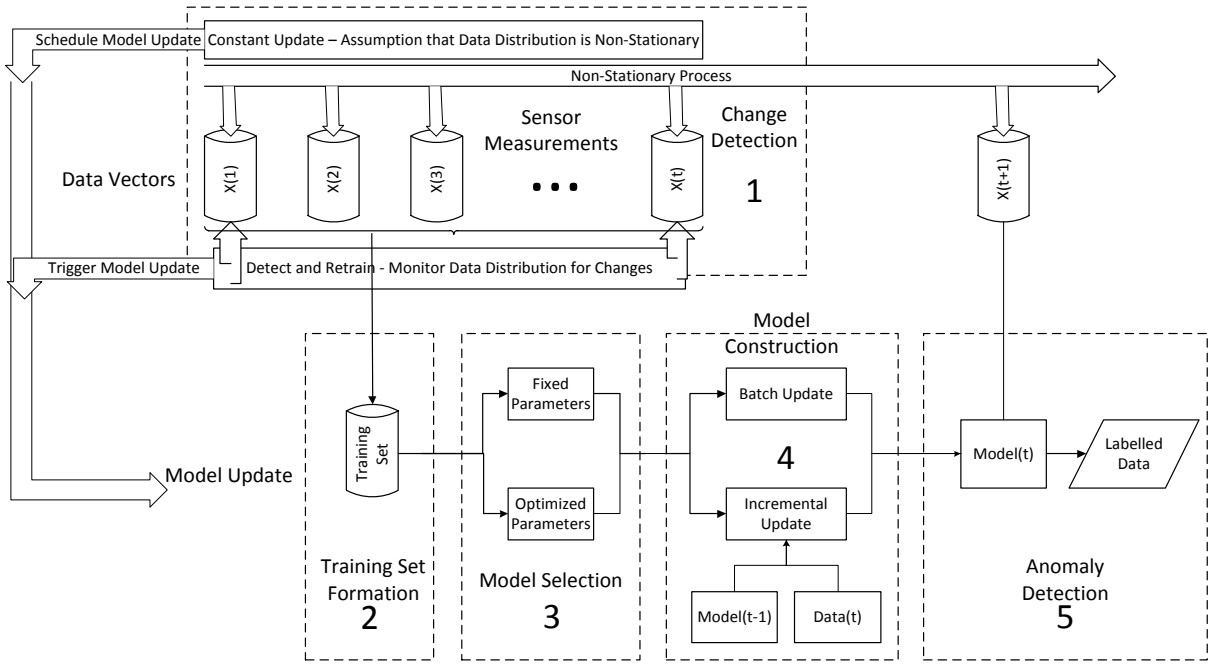


Fig. 5. Workflow detailing the different aspects of anomaly detection in a non-stationary environment in a WSN.

rates in the form of ratios can be defined. The false positive rate (FPR) is computed as the ratio of false positives to normal measurements and the true positive rate (TPR), also known as the detection rate, is the ratio of true positives to anomalous measurements.

There is a trade-off between the TPR and the FPR where adjusting a parameter, such as a threshold, to increase the TPR will result in an increase to the FPR. To examine this trade-off, a receiver operating characteristic (ROC) curve is used. A ROC curve, Fig. 6, is generated by varying a parameter, such as the anomaly rate threshold. The resulting FPR and TPR form the ROC curve. Perfect performance is achieved when there is a TPR of 1 and an FPR of 0. Performance equivalent to the random assignment of the *normal* and *anomaly* labels to the data is achieved when the TPR equals the FPR. The larger the area under the ROC curve, the better the performance of the anomaly detection algorithm.

In addition to examining the trade-off between the FPR and TPR, it is also necessary to compare the sensitivity of an anomaly detection technique to parameter selection. The area under ROC curve (AUC) [52] is used as a measurement of the performance of the scheme and is computed for a given ROC by calculating the area under the ROC curve. An AUC value of 1 indicates that the scheme has achieved 100% accuracy and an AUC value of less than 0.5 indicates that the performance is worse than the random assignment of the labels. By varying a parameter in the anomaly detection scheme, a plot of parameter versus AUC value provides a method to analyze sensitivity to parameter selection.

D. Complexity Analysis

Due to the resource constraints of a WSN it is necessary to examine the complexity of anomaly detection algorithms in order to determine computational, memory and communication complexity. A common technique used for an evaluation of this is *big O notation*. The aim is to determine the upper bound of the complexity of the algorithm. In the application domain of WSNs, it is usually used to examine how the complexity alters as the number (and possibly dimension) of the data vectors used in model construction increases.

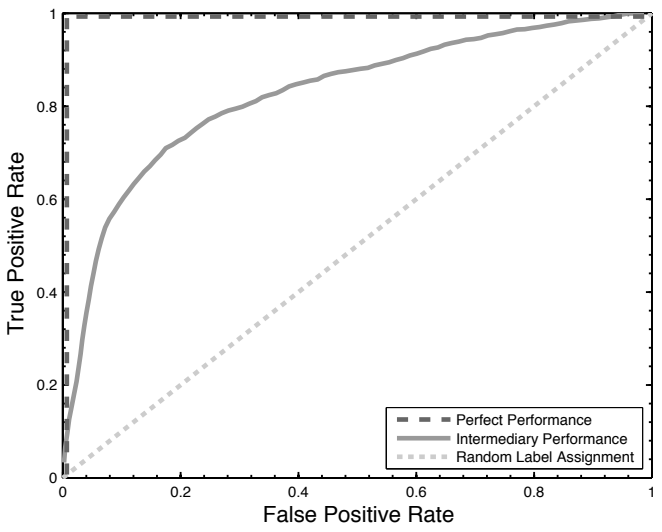


Fig. 6. ROC Curve: An illustration of the ROC space and the performance of an anomaly detector.

V. Change Detection

Due to scarce energy resources in a WSN it is important to balance the trade-off between accuracy and energy use. Issues such as a computationally expensive classifier update can be managed more effectively by determining when they should occur. For an anomaly detection algorithm to operate in a non-stationary environment, determining when change is occurring is important to avoid unnecessary updates.

Change detection can be categorized into two approaches. In one approach an assumption is made that the data distribution is non-stationary, avoiding the complexity of attempting to detect it, with updates being performed at regular intervals. In the second approach the data distribution is monitored to detect change, with an update to a model occurring when it is detected. These two approaches are termed *constant update* and *detect and retrain* [51].

A. Constant Update

One method of handling change detection is to presume that the data are generated from a non-stationary distribution and therefore to reconstruct the model at regular intervals. This simplifies matters in that no detection technique is required. However, if the timescale of the reconstruction is smaller than the timescale of the change in the non-stationary distribution, unnecessary updates are performed to the model. This could be costly in terms of computation. Conversely, if the timescale for model reconstruction is greater than the change in the data distribution, a decrease in the ability of the model to detect anomalies can occur.

Techniques for anomaly detection in WSNs often use a sliding window in order to frame data to be used as a training set. The most common manner in which it is determined when an update should occur is using *constant update* where periodically a fixed sized window advances to include an additional number of new data instances, while the corresponding number of data instances are removed from the window.

An algorithm that uses *constant update* in order to adapt to a non-stationary distribution is detailed by Zhang *et al.* [47]. The anomaly detector used in the scheme is the one-class quarter-sphere support vector machine (QSSVM) [53], a reduced computationally complex form of the one-class SVM (OC-SVM) [45]. The classifier performs anomaly detection by centering the data in feature space and enclosing normal data with a hypersphere. Fixing the hypersphere at the origin reduces the complexity so that the solution of a linear programme, rather than a quadratic, is required. The solution of the linear programme is $O(n^3)$ [54] where n is the number of data vectors in the training set. Test data vectors lying within the QSSVM are classified as normal, whereas those lying outside are labelled as anomalies. The use of kernel methods allows the classifier to derive non-linear boundaries.

The algorithm proposed by Zhang *et al.* [47] uses a sliding window approach to enable its use with non-stationary data sets. The algorithm adapts to non-stationary data by the use of two schemes involving a constant update. The first scheme uses a fixed size sliding window which progresses with each data measurement, adding the new data vector and removing the oldest data vector from the sliding window. The second scheme operates in the same manner as the first, but only updates the window every n new measurements. .

The algorithm is updated [55] using the centred hyperellipsoidal support vector machine (CESVM) [31]. The CESVM uses the Mahalanobis distance which takes into account attribute correlation whereas the QSSVM ignores attribute correlation [55]. The CESVM is resource intensive and is adapted to the resource constraints of a WSN by transforming data to a symmetric distribution in input space using the Box-Cox method [56]. The symmetric distribution is scale-invariant so data vectors are centred by subtracting the median. This operation reduces memory and computational complexity in order for it to be suitable for implementation in a WSN.

Both schemes [47], [55] aim to exploit the spatial correlation of data in order to detect anomalies in a WSN. For the QSSVM, the radius of the centred hypersphere provides a compact statistic to communicate information about the data a node is encountering. This is broadcast to spatially neighbouring nodes, and each node calculates a median radius. The two radii, local and neighbour median, are used to classify data vectors. The use of the median radius from local nodes aims to determine if a data vector would also be considered an anomaly within the context of the data in the neighbourhood. In addition, by examining the data vectors of other spatially located nodes the anomaly is classified as an event if other nodes are experiencing anomalies.

For the second scheme [55], the aim is to detect anomalies local to the node (*local anomaly*), or global to a cluster of nodes (*global anomaly*). Each node computes the CESVM, then communicates metrics describing the model; the effective radius, median and covariance matrix. From these parameters a global CESVM is formed on each node with which to identify global anomalies.

Evaluation on synthetic and GSB data sets shows that the CESVM performs better than the QSSVM on multivariate data with correlated attributes. However, the CESVM has higher computational and communication complexity than the QSSVM.

B. Detect and Retrain

A technique that aims to determine when an update to a model is required aims to *detect and retrain*. The goal is to identify a time when the data distribution has changed significantly enough to justify an update of the model. Updating a model only when required can save energy resources which is particularly important in a WSN.

Returning to the QSSVM and CESVM of Zhang *et al.* [47], [55], in addition to the constant update, a change detection scheme is outlined. The notion that identifying a period of increased anomalies signals a change point is used. We previously outlined two schemes which involve a constant update. The third scheme aims to *detect and retrain* in order to reduce the computational complexity of the algorithm. The scheme proposes reconstructing the model when new data significantly changes the model. Data vectors falling on the boundary, a border support vector (BSV), and anomalies are shown to have a significant impact on the model as they cause the constraints in the problem formulation to no longer be met. When a BSV or an anomaly is encountered, the training set is updated by adding in the new measurements since the last update, and removing the equivalent number of oldest measurements. Retraining occurs with the updated training set. As BSVs and anomalies are less common than normal data [45], this will reduce the number of model updates required. Evaluation of the technique on the IBRL [5] and GSB data sets indicate that there is a significant reduction in

the number of model updates compared to constant update methods with no reduction in performance.

The incremental hyperellipse algorithm proposed by Moshtaghi *et al.* [48], [57] makes use of change detection in order to identify changes in the data distribution in a non-stationary environment. The algorithm is an incremental update to the hyperellipsoid of Rajasegarar *et al.* [20]. Change detection occurs by monitoring the number of anomalies detected by the hyperellipse and signalling a change when the number of consecutive anomalies exceeds a threshold. Thus, identifying a change in the probability of an anomaly occurring, identifies a change in the data distribution. Evaluations showed that on synthetic data, the algorithm was able to detect change points. It is benchmarked against a model using recursive least squares (RLS) and applying a CUSUM on its residual. The incremental hyperellipse was shown to detect more change points in a non-stationary distribution in a real-world data set.

To allow the hyperellipse to adapt to changes in the data distribution without the use of sliding windows, an incremental update is detailed where new data measurements can be incorporated into the model without model reconstruction or access to the training set that originally constructed the model. An iterative update to the mean and covariance matrix allows data to be added to the model. In order to remove data measurements, a forgetting factor $0 < \lambda < 1$ is introduced. This gives a weight of λ^j to the measurement that was generated j samples previously.

A significant advantage of hyperellipses is their computational simplicity compared to other boundary techniques such as OC-SVMs. However, they are linear in nature and thus do not perform as well on non-linear data sets as kernel methods such as those derived from the OC-SVM.

VI. Model Update – Model Selection

Model selection aims to select a model from a set of (possibly infinite) candidate models that will perform optimally on unseen testing data. The parameters available to form a model from a training set vary between each technique, but can include;

- Regularization Parameter – OC-SVM
- Kernel Parameter – Kernel methods
- Subspace Dimension – Principal Component Analysis
- Anomaly Rate – Threshold techniques

There are two approaches to selecting the optimal model. In the first approach the optimal parameters for a model are estimated based on the characteristics of the training set. These are then used to construct the model. In the second approach, multiple models are constructed using different parameters, and the optimal model is chosen to be used as the anomaly detector. Different model parameters can be optimal for different data distributions. Therefore it is important that if there is a change in the data distribution, the model parameters are reoptimized to reflect this.

A. Fixed Parameters

One method of performing model selection is to fix the model parameters during deployment so that only one model is capable of being constructed from a training set. Parameters can be determined via heuristics or using specific knowledge of the domain the anomaly detection technique will be implemented in. Fixed parameters simplify model selection as the additional computation required for model selection is dispensed with.

Ensemble classifiers use multiple models in order to improve the predictive performance of an algorithm, and have gained attention in recent years. Curiac *et al.* [58] use an ensemble classifier in order to perform anomaly detection in a WSN. Five binary classifiers are used in the ensemble based system; *average based*, *autoregressive linear predictor*, *neural network*, *neural network autoregressive* and *Adaptive Neuro-Fuzzy Inference System*. The classifiers independently categorize the state of a sensor as "reliable" or "unreliable" with the final decision being determined by weighted majority voting. If required, the ensemble is used to provide an estimate of the correct measurement of a sensor affected by an anomaly.

Parameters are tuned in a training and testing phase before deployment, once the system is deployed the parameters are fixed. A detailed methodology in performing the training and testing phase is provided in order to ensure that the correct parameters are determined for a specific deployment. The methodology includes details on training the ensemble components as efficiency can only be ensured if there is diversity among the components of the ensemble [59]. To estimate the diversity of the classifiers, pairwise metrics between classifiers are used, with the mean of the metrics being used to determine the diversity of the classifier. Using the Q statistic, an overall measure of diversity is found for the ensemble, with the requirement that the value be close to 0 to ensure classifier diversity. If this condition is not met, the ensemble does not pass the testing phase and must be retrained.

Evaluation of the system occurred by training and testing the system as outlined previously in order to determine the parameters. The data sets used were either synthetic or obtained from the WSN that the system would be deployed in. Results show that the system is able to detect nodes which are producing errors, and is able to estimate the correct value. A drawback of the system is that it operates on univariate data.

Parameters can have a large impact on the generalization error of a classifier. The parameters that require setting depend on the anomaly detection technique that is being used. An example of an anomaly detection technique that uses fixed parameters is that of Rajasegarar *et al.* [31]. QSSVM [53] and CESVM are used to form a non-linear boundary in order to detect anomalies in multivariate data sets where the relationship between the attributes is not linear.

A regularization parameter, ν, is required. This parameter determines the number of data vectors that will lie outside the hypersphere or hyperellipsoid. Varying this parameter controls the trade-off between false positives and true positives. Using a ROC, it is possible to identify the value of ν that is optimal in terms of balancing the trade-off, however, this value must be set before deployment and requires ground truth labels. In addition to the regularization parameter, mapping the data into a higher dimensional space using a kernel function requires a parameter to be determined. For example, the radial basis function (RBF) kernel requires the width parameter σ. Evaluations of the QSSVM and CESVM show that the

CESVM has less sensitivity to parameter selection than the QSSVM. However, depending on the parameters chosen, there can be a large difference in the performance of the anomaly detection technique.

Distributed anomaly detection takes the form of the detection of global anomalies in a hierarchical topology using the QSSVM. After a sensor node has calculated the radius of the QSSVM, it is communicated to its parent node. The parent node combines the radii from its children with its own radii, and from this a global radius is calculated. Four strategies for calculating the global radius are proposed: *mean*, *median*, *maximum*, or *minimum*. The global radius is communicated to child nodes where it is used to classify data as normal or global anomaly.

Evaluation on the Great Duck Island (GDI) data set shows that the QSSVM [53] and CESVM have good accuracy in detecting anomalies in WSNs. However, two drawbacks to the schemes exist; computational complexity and requirement for parameter selection.

A statistical approach to anomaly detection is proposed by Zhang *et al.* [60]. Models are constructed for the detection of temporally and spatially correlated outliers in time-series data. The technique operates in an online and distributed manner. An auto-regressive moving average (ARMA) model is used to create a stationary time series and this is then used to predict future values, with actual measurements which lie outside the confidence interval being detected as outliers. A simplified version of the ARMA model was used. A reduction to $AR(p)$ means that the current observation is correlated to the previous p observations. The value of p was kept to a minimum to reduce computational complexity. The AR model on a local node was used to predict the next measurement.

The effect on performance of several fixed parameters is examined. The parameter p specifies the number of previous measurements used to predict the next measurements. It was shown that increasing p increased accuracy, however, this led to increased computational complexity of model construction. Another parameter examined was the confidence interval. The confidence interval represents a trade-off between the TPR and FPR. The use of a high confidence interval led to a high TPR and a high FPR. The use of a low confidence interval resulting in a low TPR and a low FPR. It was identified that more outliers were included in the confidence interval if the confidence level was high and these were identified as normal. From these evaluations optimal parameters were chosen.

A significant advantage of the technique is the low computational complexity. However, the algorithm operates on univariate data streams, therefore the correlation between attributes of sensor readings is not taken into account.

B. Optimized Parameters

Model selection involves selecting a model from a set of models that has a performance that can be considered optimal by some measure on a set of unseen testing data. Parameter variation provides the means to produce the set of models and parameter optimization selects the parameters that provide the optimal model.

A solution to the problem of determining the regularization parameter for the QSSVM is proposed by O'Reilly *et al.* [34]. The regularization parameter, ν, is shown to have a significant effect on the boundary of the OC-SVM and thus a significant effect on the performance of the anomaly detector. It is shown that the error rate of the classifier can be minimized by choosing an appropriate value for ν. Ratsch *et al.* [61] proposed a heuristic for determining the regularization parameter for the OC-SVM by selecting the model that separates the mean of the normal and outlier classes with the greatest distance. This is applied to the QSSVM where an online algorithm is proposed using a *golden section search* [62] to identify and track the optimal ν for a non-stationary data set. Evaluation of the scheme shows that it is able to effectively optimize the ν parameter while minimizing the number of models constructed during the model selection phase. A drawback of the scheme is the requirement for the construction of multiple models, the most computationally complex operation in the technique, in order to determine the optimal ν parameter.

Parameter estimation is performed in a Kth nearest neighbour (k-NN)-based technique proposed by Xie *et al.* [63], which optimizes two of four parameters required. The scheme uses a version of k-NN that establishes continuous hypercubes from a hyper-grid structure in feature space. Data are mapped into the hypercubes and anomalies are defined as data vectors residing outside the hypergrid. Two parameters are determined prior to implementation. The parameter b, a maximal bit length required to encode the scheme and c, a coefficient required to shift feature space into positive coordinate space. However, the algorithm is more sensitive to the two remaining parameters which are optimized. The hypercube is centred at y and has a diagonal length of the second estimated parameter, d. The parameter d is derived using the mean integrated squared error (MISE) and is shown to depend only on known constants. Estimation of the appropriate value for k is learnt from the data after the value of d has been determined. The parameter k is equivalent to the number of nearest-neighbours, but in the hypercube version it specifies the number of data vectors that must exist within a hypercube of the test vector for it to be considered normal. The value of the parameter k is derived using the analysis of the probability of an anomaly occurring.

Analysis of the complexity of the scheme determines it to be less than that of the QSSVM scheme [31] in terms of computational complexity. Evaluations performed on the IBRL data set indicate that there exists either little performance degradation compared to the standard distance-based (and more computationally complex) k-NN. Evaluations also show that there were good estimations of the optimal values for parameters k and d.

Chatzigiannakis *et al.* [33] introduce a principal component analysis (PCA)-based anomaly detection scheme that operates on multivariate data in a distributed manner. The approach fuses correlated data from multiple nodes in order to detect anomalies that span neighbouring sensors. Initially, the network of sensor nodes is divided into clusters, with nodes that have correlated data forming clusters. To form clusters, a primary node queries one (or more) hop neighbours for recent measurements. An estimation of the correlation of the data from the cluster head and the cluster node is obtained using

the correlation coefficient $R_{X,Y}$

$$R_{X,Y} = \frac{Cov(X,Y)}{S_X \cdot S_Y} \tag{2}$$

The primary node uses a predefined threshold and the correlation coefficient to select nodes to join its group. The technique has flexibility by allowing the existence of overlapping nodes between neighbouring groups and combining the individual decisions of the groups.

Once the grouping phase is complete, the nodes send data observations to the primary node. The primary node then performs analysis on the data using PCA. The principal components (PC) of the covariance matrix are sensitive to large differences between the variances of the data vectors, therefore the eigenvectors of the correlation matrix are used as the PCs. The squared prediction error (SPE) is used to detect anomalies is the data set.

Parameter optimization is used in order to determine the number of PCs that will model the data to the required accuracy. The technique of cumulative percentage of total variation [64] is often used. However, in this approach a different technique is used that is specific to the correlation matrix. Variables in the correlation matrix have unit variance, if a PC of the correlation matrix has less than unit variance it contains less information than the original variable. Therefore only PCs with variances that exceed 1 are retained.

Evaluations were performed on a data set from a real-world WSN deployment that measured meteorological data. Artificial anomalies were inserted into the dataset. Results showed that the higher the correlation of data on the nodes in a group, the better the performance. A disadvantage of the approach is the requirement to determine the detection threshold (the anomaly rate in the data set). In addition, the algorithm was shown have better performance with random anomalies compared with correlated anomalies.

VII. Model Update – Construction

Previously, techniques were detailed which determine when an anomaly detector must reconstruct a model which is used to classify data vectors. When it is determined that a model update is required, the algorithm must use the determined parameters to construct the new model. In this section the methods by which a model is constructed are detailed. Model construction involves the use of a new training set and can be categorized into batch update and incremental update.

A. Batch

Batch learning occurs when the previous data set is discarded and a new model is constructed using a new training set. Batch learning methods frame a training set with a window of data, these windows can take two forms, a *fixed* sliding window and a *weighted* sliding window.

1) Fixed Sliding Windows: Fixed sliding windows have a fixed parameter that determines how the window is formed, either containing a fixed number of data instances, or data instances from a fixed period of time. The window advances each time step, and therefore the oldest data instances leave the window, and new data instances are added. The system is simple but effective in presenting a classifier with a data set from the required period. Machine learning algorithms that assume that data are generated from a stationary distribution can be adapted to a non-stationary distribution by sliding windows. When it is determined that a model update is required, the data in the sliding window are used as the training set.

An approach using fixed sliding windows to frame the training set of a model is proposed by Xie *et al.* [65] which operates in an online and distributed manner. A fixed number of measurements in the sliding window allow adaptation to changes in the data distribution. There are three phases to the scheme; training phase, test phase and update phase. During the training phase, nodes create histograms of univariate data with bins being marked as normal or anomalous according to the histogram. The training phase includes a distributed component where cluster nodes in a WSN communicate their two-dimensional array to a clusterhead. The global normal profile is determined using the local histograms and this is then communicated back to the cluster nodes. After the training phase has been completed, the test phase determines whether a data vector is anomalous using the local histogram, the global histogram, or both.

The update phase makes use of a fixed sliding window. A test data vector is added to the sliding window with probability P_u, displacing the oldest data vector. The value of P_u is determined by how smoothly the probability density function (PDF) changes. The use of a probability test to determine whether an incoming data measurement should join the training set reduces the number of model updates that are performed. When the fixed size sliding window has a specified number of new training data vectors, an update to the model is triggered in which training phase will be repeated.

The technique proposed by Beigi *et al.* [66], [67] identifies anomalies in univariate data on local nodes. The aim is to model the current data distribution in a fixed sliding window and compare this with a baseline data distribution. The authors observe that abrupt changes in the statistical characteristics of the data indicate that anomalies are present. Histograms are used to model the data distribution as they have low complexity and can be computed quickly and updated incrementally as new data arrives. Two sliding windows are used to perform anomaly detection. One is a baseline which represents expected or historic behaviour. The second sliding window represents current data. The distribution of the current observed data is compared to the distribution of the baseline. Histograms are used to approximate the underlying data distributions of the two sliding windows of data with the difference between data distributions being measured by the Kullback-Leibler (KL) divergence metric [68]. A distance vector is created when the two data distributions are compared and anomalies are identified within this vector by identifying the maxima points. Three methods are used; *constant threshold*, *top percentage* and *maximum neighbour*. *Constant threshold* can adapt to a varying anomaly rate, where as a technique such as *top percentage* cannot. Multiple fixed-size sliding windows are used so that a multi-scale temporal analysis is performed on the data set.

Detection accuracy can be maximized by minimizing the error in generating the distribution of the data and this is performed through parameter optimization. Three schemes are outlined for the optimization of the parameters. The first scheme optimizes the bin width given the bounds on the baseline and current window size. The second scheme determines the bounds on the baseline window given the current window size. The final scheme determines the bounds on the current window given the baseline window size.

The evaluation of the scheme shows the effect of the temporal scale at which the sliding windows are set; different anomalies are detected by analyzing a data stream at different temporal scales.

2) Weighted Sliding Windows: An alternative to using fixed sliding windows is to weight the data that is used for model construction in order to give greater prominence to those items considered to be representative of the current data distribution.

Livani *et al.* [69] propose an anomaly detection scheme that uses PCA on multivariate data in a distributed manner. Nodes form clusters of nodes which are physically close and have data that are spatially correlated. One of the nodes in the cluster is designated as the cluster head. PCA is performed on the data instances on a sensor node using distributed principal component analysis (DPCA) [70]. DPCA is able to construct the global principal components on the cluster head following an intermediary calculation on the sensor node data. This reduces communication cost by reducing the amount of data that are sent to the cluster head. Fixed width clustering is then performed on data on the sensor nodes with the results being communicated to the cluster head. The cluster head then calculates the distance of each cluster from the global principal component, $\varphi(0)$, with the maximum distance, d_{max}, of all clusters from $\varphi(0)$ being determined. d_{max} is used as a threshold, with a cluster being defined as anomalous if it exceeds this distance.

An update to the global profile is calculated using the current and previous time windows. In order to give more importance to data temporally closer to the current time, a forgetting coefficient based on Ebbinghaus' forgetting curve [71] acts as a weight on the data which reduces the importance of descriptive data from older time windows.

The scheme was evaluated on the IBRL data set and it was shown that the distributed approach achieved a TPR and FPR similar to the centralized approach. The distributed approach achieved a significant reduction in communication overhead.

Another technique that handles a non-stationary distribution by weighting data vectors is that of Bettencourt *et al.* [72]. The aim of the technique is to detect events, infer missing sensor measurements, and identify anomalous sensor readings within the distributed structure of a WSN. A sensor node estimates the statistical distributions of the difference between its own measurements at different time periods and between its own measurements and that of neighbouring nodes. The estimated distributions are used to determine the likelihood of new measurements, accepting or rejecting as appropriate. This can be performed using parametric or non-parametric techniques.

In the parametric estimation of the data distribution, incremental updates to the mean and variance use a gain filter of value K_t. In order to weight all observations equally a value of $\frac{1}{t}$ is required. However, it is shown that this is not the optimal value due to correlation in the observations. As $K \rightarrow 0$ the distribution is not updated with the current measurement, as $K \rightarrow 1$ only the current measurement is used in the prediction of the data distribution. It is shown that there exists an intermediate value of K_t that minimizes the error between actual and estimated measurements. Evaluation of the algorithm shows that with non-stationary data distributions the average measurement error can be reduced with an appropriate value of K. However, the parameter K is determined prior to implementation and in a non-stationary data distribution the optimal parameter may vary over the lifetime of the WSN.

B. Incremental

A model that performs an incremental update uses the previous model in combination with the new data vectors in order to construct the new model. The incremental construction of the model can be divided into two phases. An incremental update involves incorporating new data vectors into a model. An incremental downdate involves removing data vectors from a model, the data vectors that are removed are usually the oldest data vectors. Incremental updates and downdates are advantageous as they reduce the computational complexity of building a model by reusing the old model on which computational resources have already been expended. They are designed so that the cost of adding or removing data vectors from a model is less than that of reconstructing the entire model with a batch operation.

Incremental model construction often takes the form of incorporating or removing one new data vector at a time, we term this a *step update* or a *step downdate*. Step updates and step downdates have an advantage in that there is no requirement to wait a period of time for new data to be incorporated in the model. However, updating the model as each data vector arrives can be computationally expensive.

An additional advantage of an incremental update is that it may facilitate the discarding of the data set. If the model performs one-pass learning where each data vector only needs to be accessed once, then the model contains all the data required for classification and it is not necessary to store previously seen data.

Subramaniam *et al.* [73] detail a framework which operates with an incremental step update and in a distributed manner to identify outliers in multivariate data. The technique requires only a single pass over data and has limited memory requirements. Outliers are defined based on two metrics; distance-based outliers and local metric-based outliers [74].

The aim is to find an accurate approximation of the data distribution using kernel density estimators. The algorithms for estimating the distribution are computationally efficient and implement an incremental step update by recomputing values as each new data vector arrives, allowing for an adaptation to a non-stationary data distribution without delay. A hierarchical organization of a WSN is used in a distributed learning framework with each sensor maintaining a model for the distribution of measurements it generates. A parent node takes randomly sampled subsets of child node data and combines

them to construct a global model of the distribution of the child nodes to enable detection of global outliers. Child nodes communicate outliers to the parent node for classification with the global model. The algorithm uses a fixed sliding window to adapt to non-stationary distributions, and requires an application-specific threshold to detect outliers.

Hill *et al.* [75] propose a centralized anomaly detection scheme that operates with an incremental step update on univariate data streams that have been transmitted to a central node. It uses a univariate autoregressive model of the data stream to predict the next value of a sensor measurement and a confidence interval. A measurement is considered to be an anomaly if it lies outside the interval. An assumption is made that the data stream is an order q Markov process where q is the number of previous measurements used in the model. Four data-driven methods are used to create the model to predict the region in which the next sensor reading should lie; *Naïve predictor*, *nearest cluster predictor*, *single-layer linear network* and *multilayer perceptron*. The model for the specific method is chosen using cross-validation to tune the model parameters in order to minimize the training error on the training set without over-fitting the data.

A fixed sliding window is used to adapt to non-stationary distributions with an incremental step update being performed to allow the estimation of the next data instance. Model selection for the parameters of a specific method is performed using 10-fold cross-validation on a labelled training set and thus model selection must be conducted offline. Evaluations of the technique using the FPR and false negative rate (FNR) showed that the multilayer perceptron had the best performance.

The KL divergence metric [68] is used by Li *et al.* [76] to detect anomalies in a data set. Similar to the clustering stage of Chatzigiannakis *et al.* [33], nodes form clusters based on correlation of data. Sensor nodes broadcast sensed measurements and their residual energy to h hop neighbours and the KL divergence metric is used as a measure to determine sensors with similar observations. A node will form a cluster with other nodes with similar observations to form a small network.

After clusters have been formed, anomaly detection is performed using the KL divergence metric. Nodes in a cluster transmit their data to the cluster head, which calculates and maintains a median value for the whole cluster. Divergence between the median value data set and that of a cluster node is calculated using the KL divergence metric, with anomalies being identified as those with a metric below a predefined threshold. The KL divergence metric is calculated incrementally using a step update [78]. The scheme has low computational complexity on a node, however, there is high communication cost due to the requirement for cluster nodes to transmit data vectors to the cluster head.

An incremental eigendecomposition is proposed by Chan *et al.* [77] where it is used to detect faults in WSNs. PCA is a dimensionality reduction technique that is commonly used to detect faults. However, a disadvantage of the technique is the computational complexity required in the calculation of the eigendecomposition. A subspace tracking scheme is proposed where the subspace model is updated recursively in order to incorporate new data vectors into the subspace and therefore adapt to a non-stationary data distribution. In addition, the metrics that define the anomalies in the subspace are also updated recursively.

A subspace of the data is spanned by a specified number of PCs and in this subspace outliers are identified. Two subspace tracking algorithms, PAST [79] and OPAST [80], are used in order to incrementally update the subspace online with lower computational complexity. The first method is a rank-1 modification to the eigenvectors and eigenvalues of the subspace based on the work of Abed-Meraim *et al.* [80] and having complexity $O(B^3)$, where B is the dimension of the subspace. The second method, based on the work of Yang [79], uses a deflation technique in order to perform an incremental update to the sequential estimation of the eigenvectors and eigenvalues. This has significantly lower computational complexity, but is shown to be less accurate. Fault detection occurs using a robust version of the SPE and T^2 score as they are less sensitive to the influence of outliers in the data set. To enable adaptation to non-stationary data, the thresholds are recursively updated from measurements.

Evaluation on the WSN data set Networked Aquatic Microbial Observing System (NAMOS) [81] shows that the technique offers a significant reduction in computational complexity compared with batch PCA while maintaining a similar level of accuracy to other robust subspace detection methods. The use of robust subspace tracking where outliers are removed from the training set reduces the adverse effect of anomalies. A drawback of the scheme is that data cannot be removed from the subspace.

VIII. Discussion

In this section we compare current research in the area of anomaly detection in non-stationary environments in WSNs. In addition, we discuss the complexity they add to the operation of a WSN. The shortcomings of current research are detailed and from this we recommend areas for future research.

A. Comparison of Anomaly Detection Techniques

Table I summarizes the main characteristics of the anomaly detection algorithms detailed in this survey. Table II compares the computational complexity and accuracy of the techniques surveyed. Complexity in the form of big O notation is given. Accuracy is expressed in terms of the TPR and FPR measured as a percentage and indicates the range of performance the algorithm obtains on the data set indicated. The algorithms proposed in [58], [72], [73] are excluded as they use performance metrics other than the TPR and the FPR.

The most computationally complex algorithms are those derived from the OC-SVM such as [31], [47], [55]. This is due to the solution of the linear programme required in the calculation of the boundary between normal and anomaly data vectors. In addition, there is no incremental update to reduce the computation of a new model. However, these algorithms are also the most accurate, particularly when combined with parameter optimization such as in [34]. These techniques would be best suited to environments where fewer model updates are required along with high accuracy. Hyperellipses, [48], [57], provide a less computationally complex technique to detect anomalies. These models have an incremental update

TABLE I
CLASSIFICATION AND COMPARISON. ANOMALY DETECTION ALGORITHMS SURVEYED IN THIS PAPER, AND THE TECHNIQUES USED TO ADAPT TO A NON-STATIONARY DATA DISTRIBUTIONS AS CATEGORIZED IN THE TAXONOMY (FIG. 4).

		Data	Learning	Change Detection	Model Selection	Sliding Window	Model Construction
Paper	Technique	Uni/ Multivariate	Local/Distributed /Centralized	Constant Update/ Detect & Retrain	Fixed/ Optimized	Fixed/ Weighted	Batch/ Incremental
Zhang *et al.* [47], [55]	Classification	Multivariate	Distributed	Both	Fixed	Fixed	Batch
Moshtaghi *et al.* [48], [57]	Classification	Multivariate	Local	Detect and Retrain	Optimized	Fixed	Incremental
Rajasegarar *et al.* [31]	Classification	Multivariate	Distributed	Constant Update	Fixed	Fixed	Batch
Curiac *et al.* [58]	Classification	Univariate	Distributed	Constant Update	Fixed	Fixed	Batch
Zhang *et al.* [60]	Statistical	Univariate	Distributed	–	Fixed	Fixed	Batch
Chatzigiannakis *et al.* [33]	Spectral	Multivariate	Distributed	Detect and Retrain	Optimized	Fixed	Batch
Xie *et al.* [63]	Distance	Multivariate	Distributed	Constant Update	Optimized	Fixed	Batch
O'Reilly *et al.* [34]	Classification	Multivariate	Local	Constant Update	Optimized	Fixed	Batch
Xie *et al.* [65]	Statistical	Univariate	Distributed	Constant Update	Fixed	Fixed	Batch
Beigi *et al.* [66], [67]	Statistical	Univariate	Local	Constant Update	Optimized	Fixed	Batch
Livani *et al.* [69]	Spectral	Multivariate	Distributed	Constant Update	Fixed	Weighted	Batch
Bettencourt *et al.* [72]	Statistical	Univariate	Distributed	Constant Update	Fixed	Weighted	Batch
Subramaniam *et al.* [73]	Distance	Multivariate	Distributed	Constant Update	Fixed	Fixed	Incremental
Hill *et al.* [75]	Statistical	Univariate	Centralized	Constant Update	Fixed	Fixed	Incremental
Li *et al.* [76]	Statistical	Univariate	Distributed	Constant Update	Fixed	Fixed	Incremental
Chan *et al.* [77]	Spectral	Multivariate	Local	Constant Update	Fixed	Fixed	Incremental

and change detection and would be suited to more rapidly evolving environments where more frequent model updates are required. Spectral decomposition techniques, [33], [69], [77], also provide good accuracy but have a computationally complex eigendecomposition. However, it is possible to incrementally update the eigendecomposition, reducing complexity. These techniques operate well when there is correlation between attributes which can lead to a larger dimensionality reduction. The statistical techniques, [60], [65]–[67], [72], [75], [76], also have lower complexity and are able to be updated incrementally. Some techniques also have parameter optimization. A drawback of the schemes is their operation on univariate data sets. Therefore they are applicable to applications where the required information can be learned from one attribute.

B. Change Detection

It can be seen from Table I that in most anomaly detection techniques designed for WSNs an update to the model is performed at regular intervals using a constant update. In a non-stationary environment, it is necessary to update the model to take into account the new characteristics of the data. However, in the resource constrained environment of a WSN it is essential to perform this energy costly task the minimum number of times possible. Techniques that are performing an update to a model when there is no change in the data distribution, and consequently no change in the model, cause wasted computational resources.

Another issue with the constant update approach is the setting of the temporal scale at which the update occurs. If the temporal scale of the update differs from that of the change in the data distribution, suboptimal performance occurs. If the timescale of change is greater than the update interval, model updates will be performed needlessly. If the converse occurs, and change in the data distribution occurs more frequently, the model will be constructed from data from multiple distributions.

There are approaches that are able to detect changes in a non-stationary data distribution that have been surveyed, for example [47], [48], [55], [57]. The techniques used could be exploited by other anomaly detection algorithms in order to match model updates to changes in the distribution. In addition, techniques such as CUSUM [82] and KL divergence metric [68] aim to identify when a data distribution changes. These techniques can be computationally expensive and research has been performed on reducing them to operate in more constrained environments [83]. These can be added to many anomaly detection algorithms so that a model update can then be triggered allowing a more intelligent update schedule to occur.

Change detection is particularly important if the computational complexity of model construction is high. Methods such as the QSSVM have a model construction of $O(n^3)$, a reduction in the number of models constructed can reduce the computational resource use on sensor nodes. Another resource use associated with a model update is the communication of data related to the model to other sensor nodes. Communication is the most energy resource intensive operation, therefore using change detection to reduce the number of communications required can have a significant impact on the lifetime of nodes in a WSN.

C. Model Selection

Model selection can be seen as a method of adapting to a non-stationary distribution as the optimal parameters for one distribution may differ from that of another. However, Table I illustrates that some current research omits model selection instead opting for a fixed model where the parameters are set prior to implementation. Evaluation of techniques on data sets shows that the performance can vary significantly depending on the value of the parameters chosen. Supervised machine learning problems often use cross-validation in order to optimize parameters. Unsupervised anomaly detection has no equivalent of cross-validation due to the unlabelled nature

TABLE II
COMPARISON OF THE COMPLEXITY AND ACCURACY OF VARIOUS ANOMALY DETECTION SCHEMES IN NON-STATIONARY ENVIRONMENTS. DEFINITIONS: n = NUMBER OF DATA VECTORS, $n_1 \ll n$ IN [63], d = DATA DIMENSION, q = NUMBER OF NEIGBOURS/NODES IN CLUSTER, b = NUMBER OF HISTOGRAM BINS, p = DATA CORRELATION, B = NUMBER OF PCS, [1] REAL-WORLD WSN IN CRETE, GREECE.

Citation	Scheme	Cluster Node	Cluster Head	Memory	Communication	Data Set	TPR %	FPR %
[57]	Hyperellipses	$O(nd^2)$	-	$O(d)$	-	Synthetic	$84-96$	≈ 3
[65]	Histogram	-	-	-	-	IBRL	$70-95$	$0-40$
[31]	QSSVM	$O(n^3+n^2)$	-	$O(nd+n)$	$O(1)$	GDI	≈ 90	≈ 10
[47]	Adaptive QSSVM	$O(n^3+n)$	-	$O(nd)$	$O(d)$	GSB	≈ 90	≈ 3
[55]	Adaptive CESVM	$O(nd^2)$	-	$O(nd)$	$O(d^2)$	Synthetic	≈ 93	≈ 0
[69]	Distributed PCA	$O(n^2d)$	$O(d^3log_3^q)$	$O(d^2)$	$O(qd^2)$	IBRL	$87-100$	$3-6$
[76]	Kullback-Leibler	-	Init.:$O(q(2n+6b)$ Update:$O(q)$	-	$O(n)$	IBRL	$85-95$	$2-8$
[77]	Incremental PCA	$O(B^3)$	-	-	-	NAMOS	$80-100$	$0-15$
[63]	Hyper-grid k-NN	$O(nlog(n))$	$O(n_1log(n_1))$	$O(n)$	$O(bd+log(n))$	IBRL	$75-100$	$1-9$
[60]	Statistical	$O(nd(p+q))$	-	$O(nd)$	$O(nd)$	GSB	$70-100$	$2-15$
[33]	PCA	-	-	-	-	WSN [1]	$85-95$	$0-30$
[34]	Adaptive ν QSSVM	$O(n^3+n^2)$	-	-	-	Synthetic	$85-100$	$0-4$
[75]	Statistical	-	-	-	-	Synthetic	$0-100$ (FNR)	$0-11$
[66], [67]	Statistical	$O(log_2b)+O(b)$	-	-	-	IBRL	$80-90$	$10-35$

of the data. However, certain techniques surveyed have performed parameter optimization without the use of labelled data and evaluations show that this improves performance.

A drawback of parameter optimization is that it introduces additional computational complexity on a sensor node. There is the possibility of exploiting the spatial-temporal correlation of data to distribute model selection amongst a set of nodes that share the same distribution of data measurements. Certain techniques [33], [76] include the forming of a cluster of sensor nodes with a similar data distribution. By distributing the computational complexity between nodes, the computational complexity on a single node is reduced.

Due to the unattended nature of WSNs it is necessary for sensor nodes to be able to adapt parameters if there is a change in the data distribution. Self-optimization of an anomaly detection technique can provide significant performance gains.

D. Model Update

Batch update to a model is the most common method to reconstruct a model when new data arrive that needs to be included. This is performed by the majority of the techniques surveyed. A batch update using a fixed sliding window can be computationally expensive as the previous model is discarded and a new model in its entirety is constructed.

Sliding windows are a common method to frame a training set for model construction, and to provide rudimentary adaptation to a non-stationary distribution. However, the size of the sliding window represents the accuracy/adaptation trade-off and fixed sized windows are unable to adapt this trade-off if the temporal scale of change in the non-stationary distribution alters. Most techniques surveyed in this paper use a fixed sliding window.

In addition to the accuracy/adaptation trade-off, the size of the window impacts on the anomalies that are detected. Beigi *et al.* [66] studied anomaly detection at multiple temporal resolutions and showed how different outliers, and different events, were detected at different temporal scales. By altering the size of the sliding window by factors of approximately two, different outliers were detected in the different windows. This indicates the importance of the size of the sliding window and the impact on the type of anomaly that is detected. Determining the size of the sliding window in an adaptive and optimized manner can increase the performance of the technique.

Incremental updates and downdates can provide a performance improvement as they require less computational resources in order to incorporate new data items into the model. We can view an incremental update as reusing computation conducted previously, in order to reduce the computational complexity of the update to the model. However, not all techniques are able to be reformulated to operate in an incremental/decremental manner and the computational cost does vary between different techniques.

An additional advantage of incremental/decremental techniques is that with certain techniques the model becomes the repository for the characteristics of the data and the data set can be dispensed with. This is advantageous in that there is no longer a requirement to store the data set and the algorithms can be considered to conduct only one-pass of the data to form the model. Both storage and memory resources are saved by this technique.

E. Future Directions

This survey has identified the elements that are required in order for an anomaly detection algorithm to operate in a non-stationary environment. Current research in anomaly detection in WSNs focuses on designing algorithms that create anomaly classifiers for stationary data sets, with less focus being applied to operating within a non-stationary environment. Some research has tackled the problem of a non-stationary environment and the design of algorithms that are able to adapt to changes in the data distribution.

To enable anomaly detection techniques to operate optimally in a non-stationary environment it is necessary that the algorithms are able to adapt to non-stationary distributions, are self-optimizing and update models efficiently. To this end, we recommend that future research includes the following areas:

- Application of change detection schemes that *detect and retrain* in a distributed environment in WSNs. This will allow control of the computationally complex model update. Performing a model update only when it has been determined it is necessary will increase efficiency.

- Investigation of the temporal scale of anomaly detection. Identification of the temporal scale which is required and the adaptation of the temporal scale of the anomaly detection scheme.
- Adaptive sliding windows to replace fixed sliding windows. It has been shown that the temporal scale of the sliding window is important. Determining the correct scale for the sliding window will increase the accuracy of the anomaly detector.
- Incremental updates and downdates to reduce computational complexity. Using the current model as an interim step in the computation of the next model will reduce computational complexity.
- Model selection through optimization of parameters. Determining the optimal parameters for the current training set can increase accuracy.
- Distributed model selection by exploiting the spatial-temporal correlation of data. Using nodes with a similar data distribution to share the computational complexity of determining the optimal parameters for the current data distribution.
- Incremental model selection. Extending the idea of an incremental model update to incremental model selection where the previous optimal model is used as a basis for construction of the new optimal model in order to reduce resource consumption.

IX. Conclusion

In this survey the problem of anomaly detection in wireless sensor networks in non-stationary environments was presented. A taxonomy that describes the parts of the process that allows a technique to detect and adapt to a non-stationary distribution was provided. A workflow illustrated their operation within the implementation of an anomaly detection technique in a WSN. A table provided a comparison between the surveyed anomaly detection techniques and how they adapt to a non-stationary distribution. An additional table compared the complexity and accuracy of the surveyed techniques.

Anomaly detection techniques that operate in a WSN are required to do so unattended and in an environment where the data distribution may be non-stationary. Techniques can perform more optimally in this environment if they are able to adapt to the environment they are operating in, rather than using fixed models. Techniques can improve performance if they are designed to monitor data to detect for changes in the data distribution in order to trigger a model update. When it is determined that a model update is required, optimizing parameters to the current data set, and updating models in an incremental manner can further enable a model to perform optimally and efficiently for the current data distribution.

Existing techniques that operate in an environment with a non-stationary distribution implement some of the techniques that allow adaptation. Future research should address issues outlined in the survey so that model construction occurs only when necessary and constructs the optimal model for the current data distribution. This will allow anomaly detection algorithms to minimize resource use and maximize accuracy.

Acknowledgment

The authors would like to thank the anonymous reviewers for their careful review of the manuscript and providing comments and suggestions to improve the quality.

References

[1] "SmartSantander," http://www.smartsantander.eu, 2013.
[2] F. Ingelrest, G. Barrenetxea, G. Schaefer, M. Vetterli, O. Couach, and M. Parlange, "Sensorscope: Application-specific sensor network for environmental monitoring," *ACM Trans. on Sensor Networks*, vol. 6, no. 2, pp. 17:1–17:32, Feb. 2010.
[3] A. R. Silva and M. C. Vuran, "Development of a testbed for wireless underground sensor networks," *EURASIP J. on Wireless Commun. and Networking*, vol. 2010, pp. 1–14, 2010.
[4] F. Salvadori, M. de Campos, P. S. Sausen, R. F. de Camargo, C. Gehrke, C. Rech, M. A. Spohn, and A. C. Oliveira, "Monitoring in industrial systems using wireless sensor network with dynamic power management," *IEEE Trans. Instrum. Meas.*, vol. 58, no. 9, pp. 3104–3111, Sept. 2009.
[5] "Intel Berkeley Research Lab," http://db.csail.mit.edu/labdata/labdata.html, 2013.
[6] M. Nati, A. Gluhak, H. Abangar, S. Meissner, and R. Tafazolli, "A framework for resource selection in internet of things testbeds," in *Proc. 8th Int. ICST Conf. on Testbeds and Research Infrastructures for the Develop. of Networks and Communities (TRIDENTCOM)*, Thessanoliki, Greece, Jun. 2012, pp. 224–239.
[7] X. Wang, S. Wang, and D. Bi, "Distributed visual-target-surveillance system in wireless sensor networks," *IEEE Trans. Syst., Man, Cybern. B*, vol. 39, no. 5, pp. 1134–1146, Oct. 2009.
[8] W.-T. Chen, P.-Y. Chen, W.-S. Lee, and C.-F. Huang, "Design and implementation of a real time video surveillance system with wireless sensor networks," in *Proc. IEEE 67th Veh. Technol. Conf.: VTC2008-Spring*, no. 1, Marina Bay, Singapore, May 2008, pp. 218–222.
[9] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008.
[10] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.
[11] V. Barnett and T. Lewis, *Outliers in Statistical Data*. New York: Wiley, Apr. 1994.
[12] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Detecting data anomalies in sensor networks," in *Security in Ad-hoc and Sensor Networks*, R. Beyah, J. McNair, and C. Corbett, Eds. Singapore: World Scient. Pub., Inc, Sept. 2009, pp. 231–260.
[13] ——, "Anomaly detection in wireless sensor networks," *IEEE Wireless Commun. Mag.*, vol. 15, no. 4, pp. 34–40, Aug. 2008.
[14] Y. Zhang, N. Meratnia, and P. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Commun. Surveys & Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.
[15] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. of Network and Comput. Applicat.*, vol. 34, no. 4, pp. 1302–1325, Jul. 2011.
[16] G. J. Pottie and W. J. Kaiser, "Wireless integrated network sensors," *Commun. ACM*, vol. 43, no. 5, pp. 51–58, May 2000.
[17] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan, and K. K. Saluja, "Sensor deployment strategy for target detection," in *Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applicat.*, Atlanta, GA, Sept. 2002, pp. 42–48.
[18] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. 20th Annu. Joint Conf. of the IEEE Comput. and Commun. Societies (INFOCOM)*, vol. 3, Anchorage, Alaska, Apr. 2001, pp. 1380–1387.
[19] M. C. Vuran, O. B. Akan, and I. F. Akyildiz, "Spatio-temporal correlation: Theory and applications for wireless sensor networks," *Comput. Netw.*, vol. 45, no. 3, p. 245, 2004.
[20] S. Rajasegarar, J. C. Bezdek, C. Leckie, and M. Palaniswami, "Elliptical anomalies in wireless sensor networks," *ACM Trans. on Sensor Networks*, vol. 6, no. 1, pp. 7:1–7:28, Dec. 2009.
[21] Y. Zhang, "Observing the unobservable—Distributed online outlier detection in wireless sensor networks," Ph.D. dissertation, Pervasive Systems Research Group, Faculty of Elect. Eng., Math. and Comput. Sci., Univ. Twente, Enschede, Netherlands, 2010.
[22] A. B. Sharma, L. Golubchik, and R. Govindan, "Sensor faults: Detection methods and prevalence in real-world datasets," *ACM Trans. on Sensor Networks*, vol. 6, no. 3, pp. 1–39, Jun. 2010.

[23] S. Yoon, W. Ye, J. Heidemann, B. Littlefield, and C. Shahabi, "SWATS: Wireless sensor networks for steamflood and waterflood pipeline monitoring," *IEEE Netw.*, vol. 25, no. 1, pp. 50–56, Jan./Feb. 2011.

[24] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks," *Int. J. of Distributed Sensor Networks*, vol. 2, no. 4, pp. 313–332, 2006.

[25] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. 3:1–3:42, Apr. 2007.

[26] S. Rajasegarar, A. Shilton, C. Leckie, R. Kotagiri, and M. Palaniswami, "Distributed training of multiclass conic-segmentation support vector machines on communication constrained networks," in *Proc. 6th Int. Conf. on Intelligent Sensors, Sensor Networks and Inform. Processing (ISSNIP)*, Brisbane, Australia, Dec. 2010, pp. 211–216.

[27] A. Shilton, S. Rajasegarar, and M. Palaniswami, "Combined multiclass classification and anomaly detection for large-scale wireless sensor networks," in *Proc. IEEE 8th Int. Conf. on Intelligent Sensors, Sensor Networks and Inform. Processing (ISSNIP)*, Melbourne, Australia, Apr. 2013, pp. 491–496.

[28] D. J. Hand, "Classifier technology and the illusion of progress," *Statistical Science*, vol. 21, no. 1, pp. 1–14, Feb. 2006.

[29] M. Sugiyama and M. Kawanabe, *Machine Learning in Non-Stationary Environments: Introduction to Covariate Shift Adaptation*, ser. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, May 2012.

[30] R. Elwell and R. Polikar, "Incremental learning of variable rate concept drift," in *Proc. 8th Int. Workshop Multiple Classifier Systems (MCS)*, Reykjavik, Iceland, Jun. 2009, pp. 142–151.

[31] S. Rajasegarar, C. Leckie, J. C. Bezdek, and M. Palaniswami, "Centered hyperspherical and hyperellipsoidal one-class support vector machines for anomaly detection in sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 3, pp. 518–533, Sept. 2010.

[32] M. Moshtaghi, S. Rajasegarar, C. Leckie, and S. Karunasekera, "Anomaly detection by clustering ellipsoids in wireless sensor networks," in *Proc. 5th Int. Conf. on Intelligent Sensors, Sensor Networks and Inform. Processing (ISSNIP)*, Melbourne, Australia, Dec. 2009, pp. 331–336.

[33] V. Chatzigiannakis and S. Papavassiliou, "Diagnosing anomalies and identifying faulty nodes in sensor networks," *IEEE Sensors J.*, vol. 7, no. 5, pp. 637–645, May 2007.

[34] C. O'Reilly, A. Gluhak, M. Imran, and S. Rajasegarar, "Online anomaly rate parameter tracking for anomaly detection in wireless sensor networks," in *Proc. 9th Annu. IEEE Commun. Society Conf. on Sensor, Mesh and Ad Hoc Commun. and Networks (SECON)*, Seoul, South Korea, Jun. 2012, pp. 191–199.

[35] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. of Statistical Planning and Inference*, vol. 90, no. 2, pp. 227–244, Oct. 2000.

[36] M. G. Kelly, D. J. Hand, and N. M. Adams, "The impact of changing populations on classifier performance," in *Proc. 5th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Diego, CA, Aug. 1999, pp. 367–371.

[37] G. Widmer and M. Kubat, "Effective learning in dynamic environments by explicit context tracking," in *Proc. European Conf. on Machine Learning: ECML-93*. Vienna, Austria: Springer, Apr. 1993, pp. 227–243.

[38] ——, "Learning in the presence of concept drift and hidden contexts," *Mach. Learn.*, vol. 23, pp. 69–101, 1996.

[39] R. Elwell and R. Polikar, "Incremental learning of concept drift in nonstationary environments," *IEEE Trans. Neural Netw.*, vol. 22, no. 10, pp. 1517–1531, Oct. 2011.

[40] A. Tsymbal, "The problem of concept drift: definitions and related work," Comput. Sci. Dept., Trinity College Dublin, Tech. Rep., 2004.

[41] K. O. Stanley, "Learning concept drift with a committee of decision trees," Dept. of Comput. Sci., Univ. Texas at Austin, Austin, TX, Tech. Rep. UT-AI-TR-03-302, 2003.

[42] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. 6th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Boston, MA, Aug. 2000, pp. 71–80.

[43] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proc. 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, ser. KDD '01, San Francisco, CA, 2001, pp. 97–106.

[44] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proc. 21st ACM SIGMOD-SIGACT-SIGART Symp. on Principles of Database Syst.*, Madison, WI, Jun. 2002, pp. 1–16.

[45] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[46] "SensorScope," http://sensorscope.epfl.ch/index.php/MainPage, 2013.

[47] Y. Zhang, N. Meratnia, and P. J. Havinga, "Ensuring high sensor data quality through use of online outlier detection techniques," *Int. J. of Sensor Networks*, vol. 7, no. 3, pp. 141–151, 2010.

[48] M. Moshtaghi, C. Leckie, S. Karunasekera, J. C. Bezdek, S. Rajasegarar, and M. Palaniswami, "Incremental elliptical boundary estimation for anomaly detection in wireless sensor networks," in *Proc. IEEE 11th Int. Conf. on Data Mining (ICDM)*, Vancouver, BC, Canada, Dec. 2011, pp. 467–476.

[49] L. Cohen, G. Avrahami-Bakish, M. Last, A. Kandel, and O. Kipersztok, "Real-time data mining of non-stationary data streams from sensor networks," *Inform. Fusion*, vol. 9, no. 3, pp. 344–353, Jul. 2008.

[50] U.S. National Oceanic and Atmospheric Administration, "Federal Climate Complex Global Surface Summary of Day Data," http://www.ncdc.noaa.gov/data-access/land-based-station-data/land-based-datasets.

[51] J. J. Rodríguez and L. I. Kuncheva, "Combining online classification approaches for changing environments," in *Proc. Joint IAPR Int. Workshop Structural, Syntactic, and Statistical Pattern Recognition (SSPR & SPR 2008)*, Orlando, FL, Dec. 2008, pp. 520–529.

[52] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[53] P. Laskov, C. Schäfer, I. Kotenko, and K.-R. Müller, "Intrusion detection in unlabeled data with quarter-sphere support vector machines," *PIK - Praxis der Informationsverarbeitung und Kommunikation*, vol. 27, no. 4, pp. 228–236, Dec. 2004.

[54] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. C. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Proc. IEEE Int. Conf. on Commun. (ICC)*, Glasgow, Scotland, Jun. 2007, pp. 3864–3869.

[55] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine," *Ad Hoc Networks*, pp. 1–13, Nov. 2012.

[56] K. Varmuza and P. Filzmoser, *Introduction to Multivariate Statistical Analysis in Chemometrics*. CRC Press, Feb. 2009.

[57] M. Moshtaghi, J. C. Bezdek, T. C. Havens, C. Leckie, S. Karunasekera, S. Rajasegarar, and M. Palaniswami, "Streaming analysis in wireless sensor networks," *Wireless Commun. and Mobile Computing*, 2012.

[58] D.-I. Curiac and C. Volosencu, "Ensemble based sensing anomaly detection in wireless sensor networks," *Expert Syst. with Applicat.*, vol. 39, no. 10, pp. 9087–9096, Aug. 2012.

[59] R. Polikar, "Ensemble based systems in decision making," *IEEE Circuits Syst. Mag.*, vol. 6, no. 3, pp. 21–45, 2006.

[60] Y. Zhang, N. A. S. Hamm, N. Meratnia, A. Stein, M. van de Voort, and P. J. M. Havinga, "Statistics-based outlier detection for wireless sensor networks," *Int. J. of Geographical Inform. Science*, vol. 26, no. 8, pp. 1373–1392, Aug. 2012.

[61] G. Rätsch, S. Mika, B. Schölkopf, and K.-R. Müller, "Constructing boosting algorithms from SVMs: An application to one-class classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1184–1199, Sept. 2002.

[62] J. Kiefer, "Sequential minimax search for a maximum," *Proc. Amer. Math. Soc.*, vol. 4, no. 3, pp. 502–506, Jun. 1953.

[63] M. Xie, J. Hu, S. Han, and H.-H. Chen, "Scalable hypergrid k-NN-Based online anomaly detection in wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 8, pp. 1661–1670, Aug. 2012.

[64] I. Jolliffe, *Principal Component Analysis*, 2nd ed. Berlin, Heidelberg: Springer, Oct. 2002.

[65] M. Xie, J. Hu, and B. Tian, "Histogram-based online anomaly detection in hierarchical wireless sensor networks," in *Proc. IEEE 11th Int. Conf. on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, United Kingdom, Jun. 2012, pp. 751–759.

[66] M. Beigi, S.-F. Chang, S. Ebadollahi, and D. Verma, "Multi-scale temporal segmentation and outlier detection in sensor networks," in *Proc. IEEE Int. Conf. on Multimedia and Expo (ICME)*, New York, NY, Jun./Jul. 2009, pp. 306–309.

[67] M. S. Beigi, S.-F. Chang, S. Ebadollahi, and D. C. Verma, "Anomaly detection in information streams without prior domain knowledge," *IBM J. of Research and Development*, vol. 55, no. 5, pp. 11:1–11:11, Sept./Oct. 2011.

[68] S. Kullback and R. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, pp. 79–86, 1951.

[69] M. A. Livani and M. Abadi, "Distributed PCA-based anomaly detection in wireless sensor networks," in *Proc. 5th Int. Conf. for Internet Technology and Secured Transactions (ICITST)*, London, United Kingdom, 2010, pp. 1–8.

[70] Z.-J. Bai, R. Chan, and F. T. Luk, "Principal component analysis for distributed data sets with updating," in *Proc. 6th Int. Workshop on Advanced Parallel Processing Technologies (APPT)*, Hong Kong, China, Oct. 2005, pp. 471–483.
[71] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 839–850, Apr. 2005.
[72] L. M. A. Bettencourt, A. A. Hagberg, and L. B. Larkey, "Separating the wheat from the chaff: Practical anomaly detection schemes in ecological applications of distributed sensor networks," in *Proc. 3rd IEEE Int. Conf. Distributed Computing in Sensor Systems (DCOSS)*, Santa Fe, NM, Jun. 2007, pp. 223–239.
[73] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, and D. Gunopulos, "Online outlier detection in sensor data using non-parametric models," in *Proc. 32nd Int. conf. on Very Large Data Bases*, Seoul, Korea, Sept. 2006, pp. 187–198.
[74] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in *Proc. 19th Int. Conf. on Data Eng.*, Bangalore, India, Mar. 2003, pp. 315–326.
[75] D. J. Hill and B. S. Minsker, "Anomaly detection in streaming environmental sensor data: A data-driven modeling approach," *Environmental Modelling & Software*, vol. 25, no. 9, pp. 1014–1022, Sept. 2010.
[76] G. Li and Y. Wang, "Differential Kullback-Leibler divergence based anomaly detection scheme in sensor networks," in *Proc. IEEE 12th Int. Conf. on Comput. and Inform. Technology (CIT)*, Chengdu, Sichuan, China, Oct. 2012, pp. 966–970.
[77] S. C. Chan, H. C. Wu, and K. M. Tsui, "Robust recursive eigendecomposition and subspace-based algorithms with application to fault detection in wireless sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 61, no. 6, pp. 1703–1718, Jun. 2012.
[78] M. Afgani, S. Sinanovic, and H. Haas, "Hardware implementation of a Kullback-Leibler divergence based signal anomaly detector," in *Proc. 2nd Int Symp. Applied Sciences in Biomedical and Communication Technologies (ISABEL)*, Bratislava, Slovakia, Nov. 2009, pp. 517–522.
[79] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, Jan. 1995.
[80] K. Abed-Meraim, A. Chkeif, and Y. Hua, "Fast orthonormal PAST algorithm," *IEEE Signal Process. Lett.*, vol. 7, no. 3, pp. 60–62, Mar. 2000.
[81] "Networked Aquatic Microbial Observing System (NAMOS)," http://www-robotics.usc.edu/ namos/index.html, 2013.
[82] E. S. Page, "Controlling the standard deviation by CUSUMS and warning lines," *Technometrics*, vol. 5, no. 3, pp. 307–315, Aug. 1963.
[83] C. Alippi and M. Roveri, "Just-in-time adaptive classifiers—Part I: Detecting nonstationary changes," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1145–1153, Jul. 2008.

Colin O'Reilly (M'08) received the B.Sc. degree in Mathematics from Queen Mary College, University of London and the M.Eng. degree in Telecommunications Engineering from Dublin City University.

He is currently a PhD researcher in the Centre for Communication Systems Research (CCSR) at the University of Surrey, UK. His research interests include wireless sensor networks, anomaly/outlier detection, non-stationary and distributed environments, machine learning, pattern recognition and signal processing.

Alexander Gluhak received the Dipl.-Ing.(FH) degree from the University of Applied Sciences, Offenburg, Germany, in 2002, and the Ph.D. degree from the University of Surrey, Guildford, Surrey, U.K., in 2006. He has held research positions with the Centre for Communication Systems Research (CCSR) at the University of Surrey and later with the Ericsson Ireland Research Centre.

He is currently a Senior Research Fellow at CCSR where he coordinates experimental Internet of Things (IoT) related research activities. His research interests include experimental IoT infrastructures and the application of machine learning techniques to resource constrained IoT devices to extract actionable real world knowledge and a machine understanding of human behaviour.

Muhammad Ali Imran (M'03-SM'12) received his M.Sc. (Distinction) and Ph.D. degrees from Imperial College London, UK, in 2002 and 2007, respectively. He is currently a Reader in the Centre for Communication Systems Research (CCSR) at the University of Surrey, UK. He has supervised 16 successful PhD graduates and published over 150 peer-reviewed research papers including more than 20 IEEE Journals.

He is the Principal Investigator for RCUK/EPSRC funded REDUCE project that investigated the potential of information and communications technology to reshape energy demand of end-users. In addition to this research theme, his research interests include the derivation of information theoretic performance limits, energy efficient design of cellular system and learning/self-organising techniques for optimisation of cellular system operation.

Sutharshan Rajasegarar (S'05-M'10) received the B.Sc. Engineering degree (with first class honors) in electronic and telecommunication engineering from the University of Moratuwa, Moratuwa, Sri Lanka, in 2002 and the Ph.D. degree from The University of Melbourne, Parkville, Australia, in 2009.

He is currently a Research Fellow with the Department of Electrical and Electronic Engineering, The University of Melbourne. His research interests include wireless sensor networks, anomaly/outlier detection, spatio-temporal estimations, Internet of things, machine learning, pattern recognition, signal processing, and wireless communications.