Information Fusion 33 (2017) 41–56

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Spatial anomaly detection in sensor networks using neighborhood information

Hedde HWJ Bosman[a,b,*], Giovanni Iacca[a], Arturo Tejada[c], Heinrich J. Wörtche[a], Antonio Liotta[b]

[a] INCAS[3], Dr. Nassaulaan 9, 9401HJ, Assen, The Netherlands
[b] Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600MB, Eindhoven, The Netherlands
[c] TNO, Integrated Vehicle Safety Department, 5700 AT Helmond, The Netherlands

ARTICLE INFO

Article history:
Received 29 February 2016
Revised 24 April 2016
Accepted 25 April 2016
Available online 26 April 2016

Keywords:
Anomaly detection
Sensor fusion
Sensor networks
Collaborative WSN

ABSTRACT

The field of wireless sensor networks (WSNs), embedded systems with sensing and networking capability, has now matured after a decade-long research effort and technological advances in electronics and networked systems. An important remaining challenge now is to extract meaningful information from the ever-increasing amount of sensor data collected by WSNs. In particular, there is strong interest in algorithms capable of automatic detection of patterns, events or other out-of-the order, anomalous system behavior. Data anomalies may indicate states of the system that require further analysis or prompt actions. Traditionally, anomaly detection techniques are executed in a central processing facility, which requires the collection of all measurement data at a central location, an obvious limitation for WSNs due to the high data communication costs involved. In this paper we explore the extent by which one may depart from this classical centralized paradigm, looking at decentralized anomaly detection based on unsupervised machine learning. Our aim is to detect anomalies at the sensor nodes, as opposed to centrally, to reduce energy and spectrum consumption. We study the information gain coming from aggregate neighborhood data, in comparison to performing simple, in-node anomaly detection. We evaluate the effects of neighborhood size and spatio-temporal correlation on the performance of our new neighborhood-based approach using a range of real-world network deployments and datasets. We find the conditions that make neighborhood data fusion advantageous, identifying also the cases in which this approach does not lead to detectable improvements. Improvements are linked to the diffusive properties of data (spatio-temporal correlations) but also to the type of sensors, anomalies and network topological features. Overall, when a dataset stems from a similar mixture of diffusive processes precision tends to benefit, particularly in terms of recall. Our work paves the way towards understanding how distributed data fusion methods may help managing the complexity of wireless sensor networks, for instance in massive Internet of Things scenarios.

1. Introduction

In the last decade, the vision of an internet of things (IoT) has rapidly become reality. Recent advances in technology, together with ever-decaying prices of electronic components, have made networked embedded systems ubiquitous in our life. These devices are in most cases endowed with sensing, actuating and networking capabilities and are often connected to the Internet. Noteworthy applications of these systems can be found, for instance, in home automation, automated transportation, or large scale environmental data collection [1].

While at present white goods, smart cities and buildings are being equipped with IoT technology [2], one of the earliest IoT related systems were (and are) wireless sensor networks (WSNs), with typical applications in environmental monitoring [3] and tracking of mobile agents [4]. Such applications usually require numerous sensor nodes to be deployed in remote locations. To make such systems affordable, costs are saved by reducing the quality of the sensors and the hardware resources available on each node (such

* Corresponding author.
E-mail address: sgorpi@gmail.com (H.H. Bosman).

http://dx.doi.org/10.1016/j.inffus.2016.04.007

as battery and computing elements), while the overall measurement quality of the networked system is often ensured by a high level of redundancy in measurements. For this reason, the past decade of WSN research focused mostly on optimizing resource usage [5–7].

With this body of research maturing, and the sensor technology advancing, the attention of the field is now shifting towards applications [8–11]. However, these harbor some hard theoretical problems related to the envisioned scale of the network deployments, such as the analysis of large amounts of data, stemming from, e.g., sensor networks deployed in large outdoor areas or from the many networked appliances in a smart home. The collected data is often analyzed in order to find specific information at a given point in time that is meaningful for the application to act upon. For example, seismic data could be analyzed for patterns that denote seismic activity [12], body sensor data can be analyzed to provide early health warnings [13], or vibration data could be mined for events that potentially point to a failing machine [14]. Often, such patterns or events are out of the ordinary or anomalous.

Anomaly detection can be defined as the detection of events, behaviors or patterns that are unexpected relative to a concept of what is normal [15]. A typical example is the detection of fraud in, e.g., credit card transactions or the detection of identity falsification [16]. One can also think of climate events, such as heat waves and droughts. What defines climate events as anomalous depends on multiple variables, such as location, and the proper context (drought in the Sahara desert, for instance, is not anomalous) [17]. Anomaly detection approaches are also used to detect intrusions in information systems, ever more relevant in present-day cloud computing [18].

Anomaly detection approaches is popular in applications with large central storage and processing facilities, such as those employed to process big data [19]. However, their application to lightweight systems, such as WSNs, is still limited due to the severe resource limitations posed by these systems. Limited memory and the high communication costs, for example, preclude the scenario where all WSN nodes send all information to a central facility for storage and processing [20]. To address these problems, one must either adapt to the aforementioned limitations the approaches available in the literature (which however are devised, in general, for general-purpose computers), or develop new solutions. Moreover, due to the lack of contextual information that is often not present at design time, such methods need self-adaptive mechanisms or dynamic model fitting approaches, such as machine learning techniques, to allow them to operate on data of different, unpredictable environmental conditions. Such learned models can be bootstrapped with the little information available during design time, or be learned completely unsupervised during deployment.

The decentralized nature of WSN results in measurements taken in different points in space, over time. Due to the decreasing cost of the hardware, more nodes can be deployed which results in higher quality data through redundancy. However, the measurements can contain anomalies that occur with respect to local sensors, to neighborhood information or to global information. Using anomaly detection techniques a node can, for instance, generate an initial estimate of the reliability of measurements through aggregation of local spatial neighborhood information, thus reducing the amount of data sent to a central processing facility and allowing the generation of a local and timely response to anomalies. The central processing facility could then use all the aggregated data to provide a second detection or estimation stage to improve anomaly detection accuracy, using its abundant storage and computing power resources.

In this paper, we address the following question: Can the local detection of anomalies be improved (in terms of precision or recall) by combining data from groups of spatially co-located sensor nodes? To answer this question, we devise a novel anomaly detection system based on a decentralized unsupervised online learning scheme, which incorporates local neighborhood information. We extensively evaluate this approach over a broad range of real-world network deployments and datasets from different domains. Then, in order to show the effect of the neighborhood information on the anomaly detection, we compare the performance of the framework with and without the use of neighborhood information.

The remainder of this paper is structured as follows: The next section provides a short summary of the literature related to our work. Section 3 presents our new anomaly detection approach and describes our experimental setup, while Section 4 shows and discusses our experimental results. Finally, Section 5 provides our conclusions.

2. Related work

Anomaly detection is often used in applications such as fraud detection [16], network intrusion detection [21], data centers [22], or airline safety [23]. Historical (or, *a priori*) data is used to construct a model of the normal behavior of the process (or system) under consideration, and newly arriving data is tested for fitting with the model. Patterns or behaviors that do not fit are then classified as anomalous, as fraudulent, as faulty, or simply as events that require further human analysis.

Within the research related to networked embedded devices (such as WSNs), one can often see a similar approach: Data is collected at a central point, where it is analyzed to find the anomalies. This allows, for instance, the use of multiple classifiers in an ensemble, each of which can excel in different aspects of the complex dynamics of the system under monitoring [24]. Furthermore, it allows complex transforms of multivariate time-series [25] or human reinforcement as additional detection method in, e.g., a large oceanic dataset [26].

However, central techniques have several drawbacks. The notable ones in the context of WSN systems have mainly to do with their resource usage. The wireless communication scheme also has inherent drawbacks, such as packet loss, while many detection techniques often assume reliable periodic data and, thus, have to deal with delayed packets due to retransmissions [27]. Furthermore, models learned from previously acquired data may not be suitable at any given time, and thus may require frequent model updates. Depending on the detection method used, these updates may be intrinsic and lightweight, or may require the reprocessing of all the acquired data [28].

To overcome some of these drawbacks, hybrid approaches create and update models offline that are suitable for online use in limited-resource environments. Such approaches offload the learning to a more powerful node and, thus, allow more complicated models to be learned. For example, time series are often modeled using an autoregressive moving average (ARMA) model [29]. Although, the model parameters could be estimated online, offline parameter estimation ensures that the model represents normal data, and leaves valuable computing cycles to run additional detection and classification techniques on the nodes. More complex models can only be trained offline due to resource limitations. For instance, echo state networks, a form of recurrent neural networks, can model complex time series with historical data offline. The resulting neural network can be used in WSN nodes to classify anomalies [30]. One can also think of another type of hybrid approach, where resource-limited nodes only provide basic anomaly detection methods to provide early warnings, while more complex detection methods are executed at a base station. This approach is applied, for example, in electronic health care, where WSN nodes provide early warnings based on sliding window features (such as thresholds of the mean), while a base station performs complex

processing of multiple sensor nodes, such as pattern recognition [31]. While such hybrid cases allow for a more timely response to anomalies, the need for frequent model updates, their distribution over the network and, thus, the drawback of communicating data still exists.

The wireless spectrum of a WSN is often used to collect data from the monitored process at a central location. In order to reduce communication overheads, many investigations suggest merging (fusing) data on route from a leaf node to the central node (e.g., [32–34]). For instance, in the context of anomaly detection the authors of [35] propose the use of a distributed, cluster-based anomaly detection algorithm, which locally clusters data at leaf nodes using fixed-width (fixed radius) clusters. The clusters are then merged when they are communicated towards the sink node, which has then sufficient/enough information to determine which data clusters are anomalous. This method, however, requires the data to be normalized, and will detect anomalies only globally, at the sink.

To eliminate the need for central processing/controlling entirely, a model of what is "normal" should be generated within the WSN itself. How "normal" data should look like varies depending on the context of the data under analysis and on the experience and subjectivity of the person analyzing it. Generally, normal data is a domain specific concept that requires expert consensus to be defined. It can be defined at least at two levels: normal data in the global context and normal data in the local (neighborhood context). The following subsections review both levels of anomaly detection techniques tailored to WSN, where the first section reviews global consensus approaches, and the second subsection reviews methods for a local consensus. We then conclude this review with a brief survey of online learning and detection methods recently proposed in the context of WSNs, which is particularly relevant to our proposed method.

2.1. Consensus problems

Consensus problems are "situations in which all members of some network are required to achieve some common output value using only local interactions and without access to a global coordinator" [36]. In terms of detecting anomalies, this entails a global consensus of what is "normal", such that all measurements outside of this definition are regarded as "anomalous" ones. A simple example is the task to determine the global mean and standard deviation across the WSN, with which nodes can then locally determine the anomalousness of a measurement with respect to this global consensus. However, to converge to a single global consensus, one has to account for the unreliability and limited bandwidth of wireless communications [37].

Consensus techniques can be used in combination with Kalman filters to improve the estimates of global measures [36]. Although the formal models of sensors are assumed to be known globally, and multiple communication iterations are needed to achieve a consensus usable in the Kalman filter, even if not all network members provide true readings (either due to sensor faults or intentional anomalous behavior) a consensus can still be reached, given that less than 50% of the nodes are malicious [38]. The authors prove that, if in any iteration of the consensus update neighboring node values are weighted and the extreme values are excluded from consideration, the network can still reach consensus. However, such techniques are not readily applicable to WSNs, due to their excessive communication and computational requirements, in addition to constrains on the possible network topologies.

Extra communication can be used to iteratively build a shared history of measurement data taken by all the nodes, from which a global density function (in the data space) can be estimated. With this, density-based anomaly detection can be performed [39]. By using only the messages from a local neighborhood, this approach can be adapted to perform semi-global anomaly detection. Instead of a shared history, WSN nodes can also share support vectors to train a global consensus for a support vector machine (SVM) which can then be used to categorize data in normal and anomalous classes [40].

In general, energy requirements to reach consensus are large due to their iterative approach. However, the energy usage can be somewhat optimized by choosing the appropriate transmission power for a specific network topology [41].

2.2. Local context

Methods for anomaly detection in a local context are the conceptual opposite to the afore-described centralized methods, which rely on globally shared models. In data mining, the notion of locality is often given as distance between data values (given a specific distance metric such as Euclidean distance). A data point is compared to the value of its nearest neighbors in terms of data distance [42]. However, the notion of locality can also be given in a geographical distance between the sources of the data. Many similar values (i.e., data with small distance among each other) result in a higher density, called clusters, while values that are less similar result in a lower density. Anomalies can fall outside of any cluster but, when frequently occurring, can form a cluster too. Determining if a datum is normal or anomalous compared to local neighborhood data is a challenge.

A prime example of such techniques is that of the local outlier factor (LOF) [43]. This approach compares the density around a local data point with the density around its k nearest neighbors. For each data point, a minimal radius around its values is determined such that at least k nearest neighbors are included. The ratio between the local radius and the average neighborhood radii then determines the outlier factor of a data point.

The notion of locality can, of course, also be that of geographical space. The spatial local outlier measure (SLOM) [44], is conceptually similar to LOF, but in this case the nearest neighbors are determined in geographical space. The local data is then contrasted to the trimmed mean of the neighboring data, and corrected for the 'stability' of the neighborhood, a parameter similar to variance. These and other statistical properties of local neighborhoods are described in [45], where a generalized statistical approach to obtain local statistics for further analysis (such as outlier detection) is presented.

Schubert et al. survey the above and other related and derived methods [42]. The authors unified the different approaches in a generalized framework, where the notion of locality can be interchanged between data space and geographical space. They note, however, that "making spatial outlier detection truly local remains as a possible improvement for a broad range of existing methods." Moreover, most methods target geographic information system (GIS) databases with stationary data, not time-series with evolving WSN data.

Applying these techniques to WSN is not trivial, due to the relatively high computation and the high communication cost, as surveyed in [46]. Indeed, only few of these spatial anomaly detection techniques have been applied to WSN. For instance, there are LOF-based approaches that, together with a hierarchical network structure, have been used to detect anomalous data [47,48]. Another, simplified variation of LOF is presented in [49], where the authors use the median of the data, rather than an average value, arguing that, if one wants to determine the center of a sample, the median operator is more robust to extreme (outlying) values than the mean operator. In this approach, the detected outliers are used to localize an event boundary. One common drawback of all these LOF-based methods is, however, that in order to acquire the

k-nearest neighbors one needs multiple processing iterations over the data of the network, or a more efficient hash-based aggregation technique. In both cases, these algorithms might risk exhausting the limited resources of the network very quickly.

Other approaches that target WSN use individual statistical models per neighboring node, to evaluate if the difference between the local and neighboring node is within normal range. A statistical model (e.g., mean and variance) can be learned online and applied using statistical tests [50].

2.3. Online learning and detection

As the above sections show, anomaly detection is receiving increasing attention in WSNs. Most approaches, however, are density-based, requiring all, or at least a sample, of historical data to be kept in memory. But, there are few anomaly detection approaches that actually learn models online, unsupervised, embedded in WSNs. One of those is the earlier referenced work [50], for example, where spatial correlation models are learned using mean and standard deviation statistics of differences between neighbor measurements online.

Most online learning is applied in the organization of the network, in particular in routing protocols. This includes techniques such as reinforcement learning [51], Q-learning and swarm-based methods [52]. To the best of our knowledge, few (complex) online learning methods exist that target the classification of sensed data. An example of that is an ellipsoidal SVM approach, that fits an ellipsoid to data normalized using the median of a sliding window [53].

Other examples can be found in our earlier studies, where we introduced a number of embedded algorithms for online learning of linear and non-linear models, individually [54,55], or in an ensemble [56,57]. In this paper we build upon our previous work, demonstrating, to the best of our knowledge, for the first time how neighborhood context information can be used in an automatic anomaly detection system to improve its detection capabilities in terms of precision and recall.

3. Methodology

To evaluate how neighborhood information fusion could improve the detection performance of our online anomaly detection approach, we first have to provide a context in which this approach can be applied. As mentioned earlier, our work specifically targets anomaly detection on networked embedded devices such as those used in WSNs. In such applications, the network is commonly made of a reasonably large number of nodes (tens to hundreds) with limited resources in terms of computation, memory and energy, but with several transducers to sense their environment. Within this context, in the following we assume that:

- Nodes are deployed within communication range, i.e., each node can wirelessly communicate with at least 1 neighbor.
- Nodes communications can be overheard by neighboring nodes. This can be achieved, for instance, by using Collection Tree Protocol (CTP), gossip, or other network protocols.
- Every node measures the same modalities. Although sensors do not have to be of the same make and model, their output should have the same units (such as temperature in Celsius, humidity in RH, or light in Lux).
  - Communication is reliable, that is, if a node is within communication range, it can always communicate.
  - The node positions are static.

Furthermore, we make few assumptions on the process or environment that the WSN will monitor:

- The nodes are monitoring a similar mixture of dynamic processes [58]. This mixture of processes is diffusive, i.e., overall the process behavior is correlated over space/time [59]. For example, thermodynamic processes are diffusive over space/time.
- Diffusion takes place within a measurement period.
- The process (and its diffusive properties) may change over time.
- Anomalies may occur in the process and/or in the sensor system and show a disturbance of the correlation in time or space.
  - The measurement period is smaller than the shortest time-constant of the dynamic process, such that the measurement is relevant (i.e. correlated) in this period.
  - The occurrence of anomalies is asynchronous (unrelated in time/space).

The above assumptions, the most straightforward ones indicated with the open bullets, may also be relaxed. The reliable communication, for instance, may be relaxed if the measured process dynamics are much slower than the measurement period, or when there are enough nodes in the neighborhood for aggregation. The latter also is required when nodes are mobile, to ensure a stable aggregate value. Furthermore, if the measurement period is larger than the dynamic process speed, the measurements may still contribute if the correlation is high. However, both assume that the diffusion process takes place relatively fast. If not, an additional (online) analysis can be adapted to determine the delay between positions, which can then be accounted for by buffering historic measurements [60]. Also, anomalies could occur synchronously and may be detected, if the number of anomalous nodes is the minority. However, to focus the investigation on the effect of neighborhood information, we do not relax these assumptions.

These assumptions allow us to propose that prediction-based anomaly detection methods can be improved with the use of dynamically aggregated neighboring information. This information stems from periodic updates that are sent out by neighboring nodes. The updates can be stored in neighboring nodes and, through the use of an aggregation operator, can provide extra input features that are robust to the dynamic nature of the network. In the following sections we outline our approach, and how we evaluate this proposition using a WSN simulator with topologies and sensor data traces from real-world applications.

3.1. Neighborhood aggregation

In common monitoring applications, where all measurements are forwarded to a sink node, every node periodically communicates their latest measurement. To aggregate neighborhood information, therefore, a node in the neighborhood can overhear these messages and store them locally. This push-based approach is further motivated by the claim that push messages seem more efficient (that is, have lower overhead) than pull messages [61]. However, in order to reduce data communications, an application designer can choose to broadcast messages only in the neighborhood, or with a longer period as long as the period for a modality is smaller than the shortest period in the dynamic process for a given modality. The aggregated data is then summarized through the use of an aggregation operator such as the average (or mean), the standard deviation, the median, the minimum or the maximum. While the number of neighbors may vary due to, for example, network conditions or anomalies in the data, an aggregation operator reduces these effects to a single, more stable measurement.

The measurement and anomaly detection protocol is as follows:

1. Each node measures d sensors/modalities, each with their own period p_d.
2. Each measurement is appended with a score of anomalousness based on previous local and neighborhood information.

3. Each modality can be sent separately (different packets for different sensors/modalities, because measurement periods may differ).
4. Each node buffers *recent* neighbor measurement messages per modality, with the above assumption that recent measurements are still relevant.
5. One or more aggregation operators are applied to the buffer (known anomalous measurements are excluded).
6. The aggregates are included as prediction inputs for anomaly detection.

Since, by assumption, communication is reliable and the nodes are static, then every node has at least one recent measurement from any of its neighbors, and the number of neighbors of a given node does not vary. This allows us to focus on the contribution that neighborhood information may have on the anomaly detection performance. In our experiments, the term *recent* is defined as being not older than one measurement period p_d. Due to aforementioned assumptions on the time constant of the monitored dynamic process, measurements within this recent period are assumed relevant (i.e., correlated). This is also guaranteed by the assumption that the diffusion process should be relatively fast, resulting in spatially correlated data. However, if the diffusion is slower, one may have to account for delays. For example, future work could investigate methods to automatically determine this delay in correlation, or adopt a weighted aggregation approach over a larger time period, with a weight that expresses relevance to account for correlation delays, established correlation differences or the age of the measurement. On the other hand, if the process and diffusion dynamics allow it, the definition of 'recent' can be relaxed to include multiple measurement periods to, for example, account for less reliable networks.

3.2. Neighborhood characterization

In order to evaluate the influence of the neighborhood information aggregate on the anomaly detection performance, two aspects should be considered. The first is the amount of information that can be extracted from a neighborhood. This can be estimated by the (cross) correlation between the neighborhood, the local sensors, and the aggregated neighborhood information. An alternative to correlation is a measure of spatial entropy, explained in the sequel. Establishing the amount of correlation also allows us to validate the aforementioned assumptions on the process. For example, measurements at different locations from a similar mixture of diffusive processes should be correlated because the physical phenomena of one location diffuse to another. The second aspect is the size of the neighborhood, which correlates to the network density. While one can argue that more neighboring information can result in more reliable statistics, it is reasonable to assume that neighbors at the edge of that neighborhood may measure a different (part of a) physical process that does not correlate.

Both aspects affect the correlation of the aggregated neighborhood information. That is, how well the aggregated information correlates may depend on the size of the neighborhood, and on the aggregation operator chosen. Furthermore, the latter may prove more or less robust to variations in neighborhood size. Thus, we first investigate the correlation of the neighborhood and of the aggregation operators applied to that neighborhood for varying neighborhood sizes. Then an aggregation operator is chosen that correlates best across those sizes. Finally, using the chosen aggregation operator, the influence of neighborhood size on the anomaly detection performance is investigated.

We use the Pearson correlation coefficient as a measure of the linear correlation between two variables **a** and **b** [62]. Its value ranges from –1 to 1, where 1 is a perfect positive correlation, –1 is a perfect negative correlation, and 0 means no correlation. A low correlation would be $|r| < 0.25$, a high correlation means $|r| > 0.75$. For a given sample of size n for both **a** and **b**, the correlation coefficient r can be expressed as:

$$r = \mathrm{corr}(\mathbf{a}, \mathbf{b}) = \frac{\sum_{i=1}^{n}(a_i - \bar{a})(b_i - \bar{b})}{\sqrt{\sum_{i=1}^{s}(a_i - \bar{a})^2}\sqrt{\sum_{i=1}^{s}(b_i - \bar{b})^2}}$$

Since negative correlation is also correlation that contributes to the model, we take the absolute correlation $|r|$. In the sequel, correlations between local sensors and neighborhood information are averaged over each node in a deployment scenario to account for differences in node neighborhoods, which depends on the topology in the scenario. In order to account for bias in averaging of correlations, Fisher's Z-transform, $z = Z(r) = \mathrm{arctanh}(r)$, should be applied before averaging, and the inverse transform, $r = Z^{-1}(z) = \tanh(z)$, on the result [63].

The correlation coefficients are averaged over all nodes in a scenario and stored in a matrix, which can be graphically depicted as a heat map (i.e., a correlation map). In the following, the creating of this matrix is explained. We refer to the neighborhood of node i as $N(i)$. Sensor modalities are referred to with subscript indexes s and m. Then, measurement time-series data of node i for sensor m are referred to as x_m^i. For the buffered neighborhood data of node $j \in N(i)$ the data are referred to as $x_m^{i,j}$. The set of neighborhood measurement time-series for sensor s is $\{x_s^{i,j} : j \in N(i)\}$, for brevity sometimes referred to as X_s^i, and we can aggregate those per time-instance using an operator OP, such as the mean, resulting in a single time-series. The correlation coefficients r are calculated for each pair **a** and **b** of measurement time-series from local sensors (in a given node i, e.g., $\mathrm{corr}(x_0^i, x_1^i)$), local sensors and neighborhood aggregates (e.g. $\mathrm{corr}(x_0^i, \mathrm{OP}(\{x_m^{i,j} : j \in N(i)\}))$ where i is the node under investigation), local sensors and sensors of neighboring nodes (e.g. $\mathrm{corr}(x_0^i, x_0^{i,j})$ where i is the node under investigation and $j \in N(i)$ a neighbor), pairs of neighborhood nodes (e.g., $\mathrm{corr}(x_0^{i,j}, x_0^{i,k})$ where $j, k \in N(i)$ are two neighbors of i). Similarly, we compare neighborhood aggregate time-series to the neighboring node measurements and neighborhood aggregate time-series to other neighborhood aggregate time-series of different operator types (e.g., mean and median) in order to explore how well the operator correlates with (summarizes) the neighborhood and if it differs from other operators. These are then averaged for all nodes in a given scenario using Fisher's Z-transform.

This process is summarized in Algorithm 1, where the keys of the map M are strings and, as such, they are indicated with double quotes (" ") to distinguish them from numerical values. Moreover, because the correlation coefficient between **a** and **b** is the same as the correlation between **b** and **a**, the matrix is symmetric. The diagonal of this matrix should be one, as the correlation coefficient of a signal with itself is one. However, in our matrix, we also compare the correlation among the neighbors of a node, which results in a less than one correlation because we are not comparing a neighboring signal with itself, but with another neighbor's signal of the same sensor. Using the correlation map, we can see stronger correlations having a darker color, which allows us to visually examine the relevance of neighborhood aggregates to local sensor values, and compare them to the raw correlation per neighbor.

The correlation coefficients between local sensors of a node shows how well they correlate and, thus, how much information can be obtained locally. This can be compared to how well the neighboring sensors correlate, which gives an indication of how well aggregated neighborhood information should correlate. Most important, the average correlation between pairs of neighboring nodes can be compared to the correlation between a local sensor and the neighborhood aggregate, to form an indication of the ag-

Algorithm 1 Correlation map creation

correlation map $M \leftarrow 0$
for each node i in scenario **do**
  for each local sensor pair m, s of node i **do**
    $M[\text{“}x_m\text{”}, \text{“}x_s\text{”}]$ += $Z(|\text{corr}(x_m^i, x_s^i)|)$
  $l \leftarrow$ number of neighbors in $N(i)$
  for each sensor pair m, s **do**
    for each aggregate operator OP_a **do**
      $X_m^i \leftarrow \{x_m^{i,j} : j \in N(i)\}$
      $X_s^i \leftarrow \{x_s^{i,j} : j \in N(i)\}$
      $M[\text{“}x_m\text{”}, \text{“OP}_a(X_s)\text{”}]$ += $Z(|\text{corr}(x_m^i, \text{OP}_a(X_m^i))|)$
      for each aggregate operator $\text{OP}_b : \text{OP}_b \neq \text{OP}_a$ **do**
        $M[\text{“OP}_a(X_m)\text{”}, \text{“OP}_b(X_s)\text{”}]$
          += $Z(|\text{corr}(\text{OP}_a(X_m^i), \text{OP}_b(X_s^i))|)$
      for each neighbor $j \in N(i)$ **do**
        $M[\text{“OP}_a(X_m)\text{”}, \text{“}x_s\text{”}]$
          += $Z(|\text{corr}(\text{OP}_a(X_m^i), x_s^{i,j})|)/l$
    for each neighbor $j \in N(i)$ **do**
      $M[\text{“}x_m\text{”}, \text{“}X_s\text{”}]$+= $Z(|\text{corr}(x_m^i, x_s^{i,j})|)/l$
  $p \leftarrow$ number of neighbor pairs in $N(i)$
  for each pair of neighbors j, k in $N(i)$ **do**
    for each sensor m of node j **do**
      for each sensor s of node k **do**
        $M[\text{“}X_m\text{”}, \text{“}X_s\text{”}]$ += $Z(|\text{corr}(x_m^{i,j}, x_s^{i,k})|)/p$
$M = Z^{-1}(M/(\text{number of nodes in scenario}))$

gregation operator's capability to represent the neighborhood information. In a later experiment, then, we test the intuition that more correlated information contributes more to the anomaly detection performance.

As an alternative to correlation, we use the spatial entropy [64], a measure of complexity of spatial systems defined as:

$$H = -\sum_i p(e_i)\Delta e_i \log(p(e_i)\Delta e_i).$$

This value gives an entropy figure based on the probability $p(e_i)$ of an event in an area Δe_i. The probability in this case is the chance of an anomaly occurring at a specific node i, and is estimated using labeled events from our data sets. The exact area that a node senses, however, is unknown. But, we do know the node positions. With those, a Delaunay triangulation and a Dirichlet tessellation (or, Voronoi diagram) can be constructed to create an estimated area (around the node positions) that a node can sense [65]. To calculate Δe_i, then, we can then take either 1/3 of the total area of the three Delaunay triangles emanating from a node position e_i, or the cell size from the Dirichlet tessellation, as seen in Fig. 3c. The resulting values represent the information that can be gained from neighborhood information, where lower spatial entropy values imply more gain. Both the spatial entropy and the cross correlation measures can give us an indication of the validity of the assumption that the process or environment consists of a diffusive mixture of dynamic processes, resulting in correlated behavior over space and time.

The influence of the size of the neighborhood depends on either the radio communication range, or the density of the deployment. In order to simulate a change on these, either the radio range or the positions of nodes can be changed. Since the latter are known for the real-world datasets used in this study, but the exact radio parameters are not, we opt to change the communication range by changing the parameters of the radio model in the simulator. The radio propagation model is a Log-distance path loss model, which predicts the path loss over distance to have a logarithmic decay, with optional Gaussian noise to simulate interference [66].

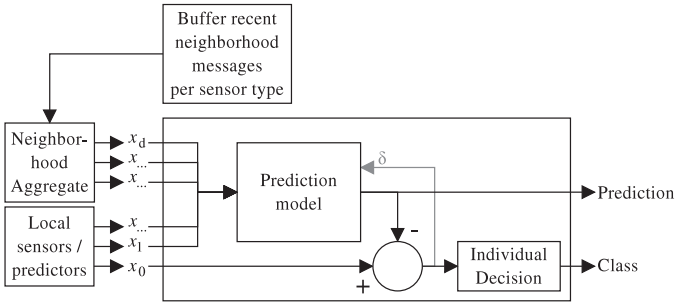

Fig. 1. Structure of a multi-dimensional time-series classifier. The difference between a prediction, based on inputs $x_1, \ldots, x_d$, and the current measurement in time-series x_0 is classified.

The static parameters of the model are the unit distance $D_0 = 1.0$, the path loss at this reference distance $PL_{D_0} = 55.4$, and the noise floor is –106.0 dB. Since we assume reliable communications, we do not add a noise component to the radio model. The main parameter is the path-loss exponent (PLE), which dictates the decay of signal strength over distance. That is, higher values of PLE result in higher path loss and thus a smaller radio communication range, whereas low values result in less path loss and a larger communication range. Therefore, we vary the PLE, effectively changing the number of neighbors in the neighborhood, and measure the result on the change in anomaly detection performance, compared to the same classifiers without neighboring information.

3.3. Embedded online anomaly detection

In a limited resource and limited precision environment, we can use incremental learning techniques to learn (or fit) linear and nonlinear relationships between measurements from different sensors or historical measurements. Incremental (or sequential) learning techniques allow a model to be updated when new data becomes available, and do not require a large historic dataset to be kept in memory. Thus, the main resource usage results from computations and the models. The predictions of these models are then compared to the measured value, and the difference is analyzed to detect anomalies. A graphical flow of this approach is shown in Fig. 1.

In particular, we use recursive least squares (RLS) to learn linear models, and the online sequential extreme learning machine (OS-ELM) approach to train a single-layer feed-forward neural network (SLFN) [54–56]. The latter approach randomly sets input weights and biases, requiring only the output weights to be learned, which can be done with RLS. This extreme learning machine (ELM) approach was demonstrated by Huang et al. [67], similar to random vector functional-link neural networks [68], to perform on par with other machine learning methods, such as support vector machines, given enough hidden neurons. Furthermore, we also include polynomial function approximation (FA) and sliding window mean prediction methods as single time-series predictors [57].

The single time-series predictors make use of windows of recent historical measurements. From this window we extract the average (mean), but also fit a polynomial function, that models the trend of the data. Using the fitted function, then, we can extrapolate the trend to predict future measurements. Due to the limited (16 bit fixed-point) precision available we opt for a linear function fit. The function approximation is done incrementally, allowing for an embedded implementation, using a method called Swift-Seg, that has a complexity and memory footprint in the order of the buffer length and polynomial degree [69]. The aforementioned local predictions are not influenced by neighborhood information.

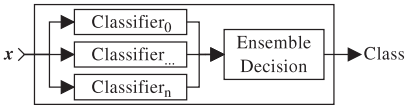

Fig. 2. Structure of an ensemble of classifiers. The final decision on *class* is based upon the outputs of the different classifiers.

Next to the memory limitations (resulting in small models), these methods have to be adapted to a limited precision environment (resulting in buffer under- and overflows and stability issues). In general, such issues can be identified by analyzing the math operations in the algorithm. In particular, for the algorithms above, we have identified the following major issues to be addressed. First, the inputs have to be scaled such that the expected minimum and maximum values do not (often) run into the boundaries of the limited precision and, if they do, should saturate instead of rolling over. Next, the RLS and OS-ELM methods may suffer from instability issues due to the limited precision. In earlier work, we showed that the methods may be stabilized, among others by not rounding math operations, and correcting the inverse auto-correlation matrix [54,55]. Finally, the FA methods uses variables that accumulate values and, thus, may run into precision boundaries. In such case, the model is re-initialized with the previously buffered values. Note that higher degree polynomials require a higher number of accumulating variables and run a higher risk of fixed-precision overflow. Hence, here we will limit our analysis to first degree polynomials. With these adaptations, the methods run stable in limited precision environments.

The RLS and OS-ELM learned models are used to make predictions based on input features. In the methods presented in our previous work, hereafter called the *stand-alone* or SA methods, these features were either from only local sensor data, or the fusion of local sensor data with local predictions from other models. The latter (referred to as RLS fusion or OS-ELM fusion) included for each modality the raw sensor data, the previous measurement, the function approximation prediction, and the window mean. In previous work, this fusion of raw data with other local predictions showed a clear improvement in the precision of anomaly detection [57].

In this work, we replace two of the input features of the stand-alone methods by neighborhood aggregated measurements. Specifically, by replacing the local input features of previous measurement and of window mean by two neighborhood aggregates in the classifiers, the change in detection performance can be evaluated with the metrics described below. Due to the assumption of a similar mixture of diffusive processes, we expect that including these aggregates the anomaly detection performance will increase the RLS and OS-ELM fusion classifier performance. Moreover, the ensembles (that combine multiple classifiers as shown in Fig. 2) are also expected to be positively affected.

3.4. Deployment scenarios

In order to evaluate the benefit of neighborhood information in the embedded online anomaly detection methods, described above, several real-world WSN monitoring scenarios are defined. These include the traces of sensor data and the related network topology. To make use of the known topologies, we use the radio parameters defined in the last paragraph of Section 3.2, with varying PLE to emulate different network densities, the effect of which can be seen in Fig. 3. The scenarios are then used in the TinyOS TOSSIM simulator [70], which emulates a WSN on radio layer and up.

The topology and datasets are derived from three existing applications. The first is the SensorScope Grand St. Bernard scenario[1],

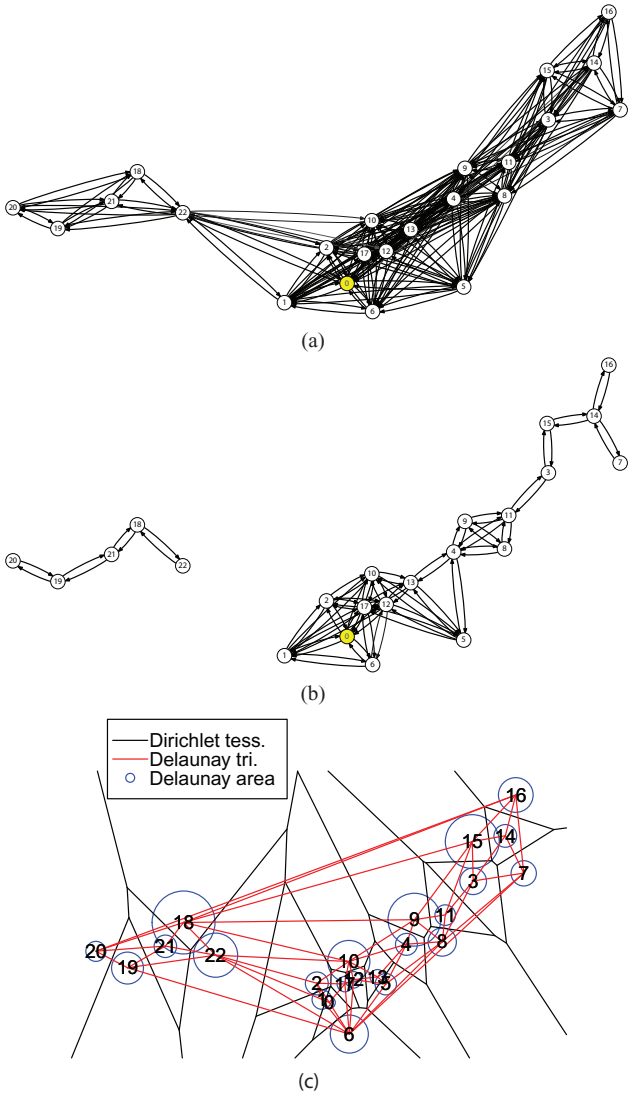

Fig. 3. SensorScope Grand St. Bernard network topology. (a) A dense network simulation with PLE = 4.0, (b) A sparse, disconnected, network simulation with PLE = 6.2, (c) The Dirichlet tessellation, Delaunay triangulation, and node area based upon the triangulation. Note that even though the network in (b) is disconnected, each node can communicate with a neighbor.

in the following referred to as GSB, which contains meteorological data (ambient and surface temperature, humidity, soil moisture, solar radiation, and watermark) collected for one and a half month in 2007 at the Grand St. Bernard pass located between Switzerland and Italy[2]. The second dataset originates from the Intel Berkeley Research Lab[3], henceforth called Intel Lab, where 54 sensors measured temperature, humidity, and light in an indoor office setting. Similarly, the third dataset originates from our own testbed located indoor, referred to as Indoor WSN, spanning several offices. All 3 datasets have been labeled by a semi-automated method, as described in [57], where rule-based labeling was checked manually. During the labeling, we distinguished four types of anomalies,

[1] http://lcav.epfl.ch/page-86035-en.html

[2] Note that, since we run the anomaly detection methods on WSN motes with limited resources, including memory, the algorithms were constrained to handle at most three sensors at a time. Therefore, we split the sensors from the GSB data into two sets: Those sensors that are *temperature* related (ambient and surface temperature, and relative humidity), and *humidity* related (relative humidity, soil moisture and watermark).

[3] http://db.csail.mit.edu/labdata/labdata.html

Table 1
The used datasets and their properties. The dimension (dim.) are in meters, the percentage of anomalies (%anom) is based on the total number of samples (#smp).

Dataset/topology	Dim.	#node	#smp	%anom
GSB	56 x 28	23	0.58 M	5.1%
Intel Lab	30 x 40	54	2.3 M	19.9%
Indoor WSN	17 x 17	19	0.8 M	2.7%

also indicated in literature (e.g., [71]): spike (short high intensity spikes), noise (increased variance over a period of time), constant (a constant value over time) and drift (an offset over time) anomalies. Table 1 lists these real-world scenarios with dataset and topology properties.

3.5. Evaluation metrics

In order to evaluate the effect of including neighborhood information in the anomaly detection methods, we compare the performance of our methods with and without the neighborhood information using several metrics. The anomaly detection performance is measured using a confusion matrix, and measures based thereupon [72]. The confusion matrix lists a count of True Positives (TP), False Positives (FP), False Negatives (FN) and True Negatives (TN). A TP occurs when an anomaly is present in the data and the node detects this, a FP occurs when there is no anomaly in the data, but the node concludes there is one. Similarly, a TN indicates that there was no anomaly in the data and the node indeed concludes there is no anomaly, while a FN shows the number of times a node did not detect an anomaly when one was present.

From these confusion matrix counts, we can further derive the following metrics:

- Precision, expressed as $TP/(TP+FP)$, shows the ratio of true detections over all detections.
- Recall, expressed as $TP/(TP+FN)$, shows the ratio of existing anomalies in the data that are detected.
- F-measure, expressed as $(2 \times \text{precision} \times \text{recall})/(\text{precision} + \text{recall})$ gives a single average measure of performance.

Depending on the goal of the application, one can opt to focus on only a subset of these metrics. For instance, when an operator should not be overloaded with false positive detections (false alarms), a higher precision is required, such that a detection is more likely to be a true anomaly. When, for example, an offline system with abundant resources complements the detection methods by improving on precision, a higher recall is beneficial, such that more anomalies are found while false positive detections can be filtered by the offline system.

4. Results

To evaluate the influence of neighborhood information on anomaly detection performance, we first characterize the neighborhood by analyzing the possible relevance of neighboring information using correlation coefficients, by examining which aggregation operators may be best to reduce the neighborhood to a single representative value for local processing and by analyzing the influence of the neighborhood size. That is, we try to answer the question: Will including more neighbors make the anomaly detection perform better? Finally, we analyze the detection performance for an optimal neighborhood size, and for a less than optimal size.

4.1. Neighborhood characterization

In order to minimize the effect that neighborhood size may have on our choice of aggregation operator, we analyzed correlation maps from the same dataset with different radio model parameters. For example, Fig. 4 shows the correlation maps of increasing neighborhood size for the Grand St. Bernard humidity scenario. The effect of the neighborhood size can be seen immediately, by the overall darker colors of the correlation map of the denser network, showing a higher correlation.

The correlation coefficients in the bottom left of these maps in Fig. 4 show that local sensor 2 and 3 are more correlated than sensor 1 and 2 or sensor 1 and 3. This pattern repeats in the top right of the map, showing that the values of sensors 2 and 3 between neighbors are more correlated than the other sensors in the neighborhood, albeit less than the correlation between the local sensors. Moreover, we can see that the neighborhood aggregates correlate better between aggregates of the same sensor than others. Especially the mean and median are highly correlated, while the standard deviation has low correlation throughout. The mean and median also correlate to the minimum and maximum values in the neighborhood, but to a lesser extent than the correlation between mean and median. Overall, as neighborhood size increases, the aggregate operator correlations increase and the mean and median seem a reliable choice.

In Fig. 5 two different scenarios are depicted, namely the Intel Lab and the Indoor WSN testbed datasets. Both have similar environments (indoor offices), and have the same set of sensors (temperature, humidity and light), but the Intel Lab dataset shows much higher correlation throughout. Interestingly, the dataset similarities also show in the correlation patterns. That is, sensors 1 and 2 (temperature and humidity) have higher correlation between each other than the light measurements have with any of them. However, there are large differences due to the environment. The main finding is that here, too, mean and median of a neighborhood, for a specific sensor type, are highly correlated to the local sensors. The minimum and maximum show a slightly lower correlation and more variation. This indicates that both mean and median operators should be a good choice to aggregate neighboring information.

Lastly, to characterize the datasets, we analyze their average cross-correlation and spatial entropy. In Table 2, the cross-correlation value is the average cross-correlation over the whole dataset. The table shows that the Intel Lab dataset has the highest cross-correlation and the lowest spatial entropy (with both area measures). Therefore, we expect the information gain of neighborhood information to be high. While the second highest cross-correlation value is from the Indoor WSN, the higher spatial entropy values might indicate that the information gain of spatial neighborhoods might be less. Both GSB datasets have relatively low cross-correlation and higher spatial entropy. Therefore, the neighborhood information gain will most likely be little. Overall, we expect that the Intel Lab dataset will gain the most from neighborhood information.

Table 2
The average cross-correlation and spatial entropy of the datasets. The spatial entropy [64] is based on the chance of anomalous measurements per area, where the area is based on the Delaunay triangulation or Dirichlet tessellation over known node positions.

Dataset	Cross-corr.	Spatial entropy	
		Delaunay	Dirichlet
GSB Humidity	0.131	0.159	0.211
GSB Temperature	0.191	0.255	0.269
Intel Lab	0.615	0.141	0.143
Indoor WSN	0.261	0.296	0.268

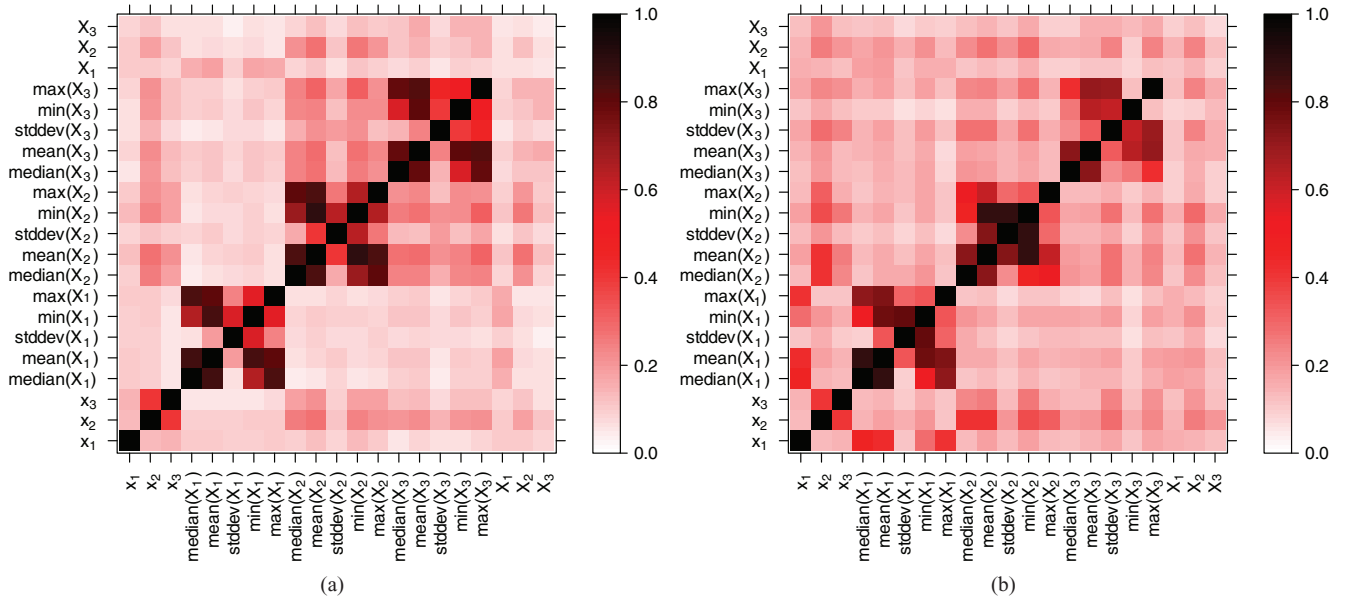

Fig. 4. A denser connected network results in better correlation with neighborhood aggregates. Furthermore, it shows different sensor modalities have different correlations with their neighborhood and with other sensors (sensor 2 and 3 are more correlated than sensor 1 and 2). The correlation maps show the average correlation of the Grand St. Bernard humidity related sensors, with (a) a sparse network (PLE = 6.2), and (b) a denser network (PLE = 4.0).

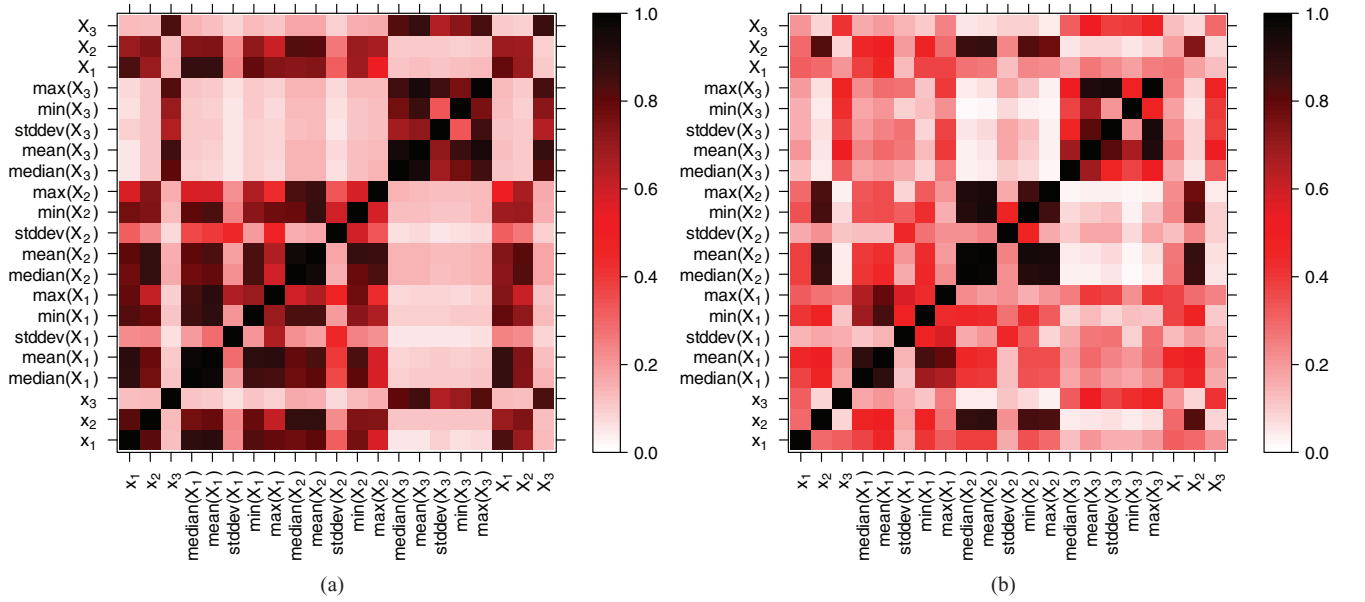

Fig. 5. With the same set of sensors but different environments the correlation between sensors and their neighborhood aggregates shows similar patterns. However, due to environmental differences, the average correlations are lower in the Indoor WSN dataset (b). The correlation maps stem from (a) The Intel dataset (PLE = 4.0) and (b) the Indoor WSN dataset (PLE = 4.6).

4.2. Neighborhood size

Having established that the mean and median are suitable choices for aggregation operators, and that the Intel Lab dataset fulfills the assumption of a similar mixture of diffusive processes more than the other datasets, the influence of neighborhood size can be analyzed. By changing the PLE the global average number of neighbors changes in the network, as seen for each dataset in Fig. 6. Due to the limited memory size available in the WSN nodes, the number of neighborhood messages stored is 20, which shows as the maximum number of neighbors in the figure. Thus, the size can be analyzed on the global network, but also per number of neighbors that individual nodes have. For the sake of brevity, in the remainder of this section we show only the results of the RLS-fusion classifier. The results of other affected classifiers result in equal findings.

4.2.1. Network average number of neighbors

We first look at the average number of neighbors given a PLE setting. Fig. 7 shows the change in F-measure for the RLS-fusion detector, as a result of replacing some input features with neighborhood aggregates, plotted against the average number of neighbors. In Appendix A we show the change in F-measure for given PLE settings, from which this figure is derived. To get a better insight in the trend, polynomials of first to fourth order are regressed to this data. The highest order polynomial that has significant improvements over lower-order polynomials according to the ANOVA test with p-value = 0.05 is displayed.

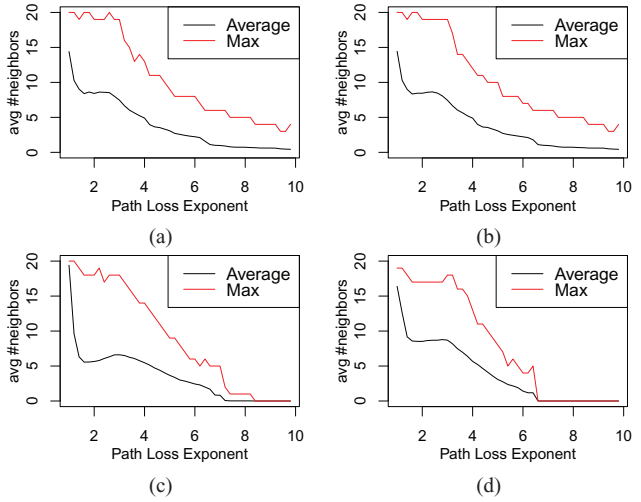

Fig. 6. The number of neighbors decreases when the radio range decreases. The maximum of 16 is the result of the neighborhood buffer size. The figures show PLE vs average number of neighbors for the GSB topology in the case of (a) humidity and (b) temperature, which should be equal. For the Intel Lab topology (c) and for the Indoor WSN topology (d) the ratios are different.

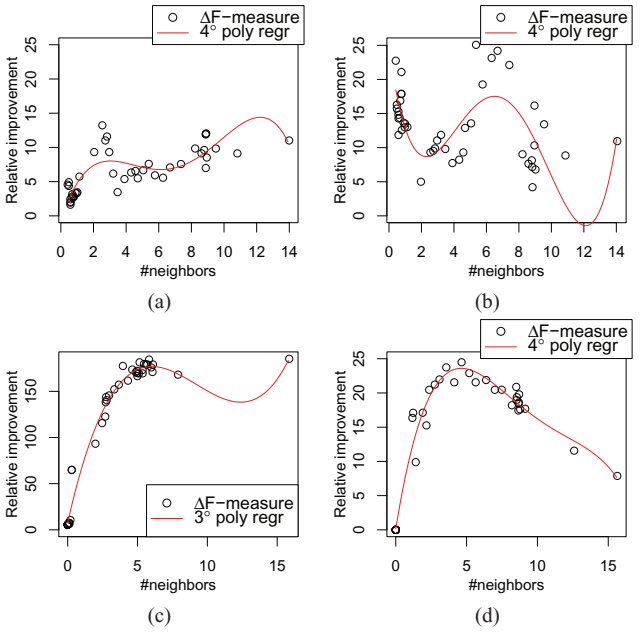

Fig. 7. The optimal number of average neighbors depends on topology and sensor modality. The plots show average number of neighbors vs the relative change in F-Measure of the RLS-fusion classifier. The datasets are (a) GSB humidity, (b) GSB temperature, (c) Intel Lab (note the different y scale), and (d) Indoor WSN.

From these figures, we can already see that for all the datasets the inclusion of neighborhood information does seem to improve the F-measure performance, albeit to a varying degree. The cases where the PLE is too large (and thus radio range too small) to allow any communications mostly result in zero improvement. Moreover, in most cases there is an optimum average number of neighbors (and thus an optimum radio range). However, the optimal radio range depends not only on the topology but also on the dataset. For example, the GSB temperature and humidity datasets in Figs. 7a and 7 b share the same topology, but have different optima. On the other hand, the Indoor WSN dataset shows a clear peak around an average of 4 neighbors, and the optimum of the Intel Lab dataset lies around an average number of neighbors of five. The Intel Lab dataset also shows the highest relative improvement in F-measure of over 150%, which can be contributed to the dataset characteristic of being highly correlated. In the future, the framework may benefit from adaptive transmission power control, such that an optimal number of neighbors can be chosen for a deployment.

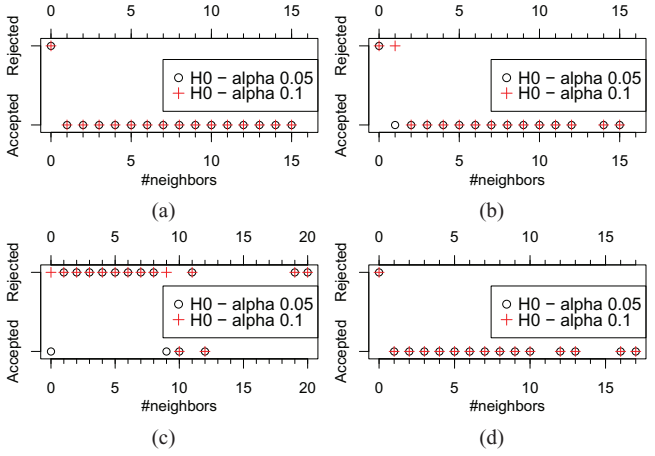

Fig. 8. Precision does not improve except with the Intel Lab dataset. The plots show if the *precision* statistically significantly improved, and thus H0 is rejected. The datasets are (a) GSB humidity, (b) GSB temperature, (c) Intel Lab, and (d) Indoor WSN.

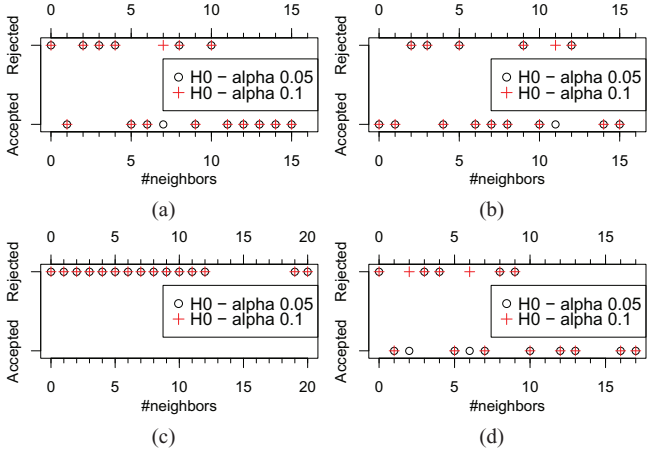

Fig. 9. Recall does improve depending on the number of neighbors and dataset used. The plots show if the *recall* statistically significantly improved, and thus H0 is rejected. The datasets are (a) GSB humidity, (b) GSB temperature, (c) Intel Lab, and (d) Indoor WSN.

4.2.2. Exact number of neighbors

The above analysis was made using the average number of neighbors a network had at a given PLE setting. To better understand the exact influence of the neighborhood size, we now analyze the results of the RLS fusion classifier per number of neighbors, over the whole PLE range (from 1.0 to 10.0). In this case, we study the effect of replacing certain input features with the neighborhood aggregates on the precision and recall separately. Both the results of the stand-alone methods and the methods with neighborhood information form two distributions. Using the Kolmogorov–Smirnov test [73], these distributions can be compared and tested if the new recall and precision measurements statistically significantly improve over the stand-alone methods.

Figs. 8 and 9 show the results of the Kolmogorov–Smirnov test for precision and recall respectively. Our null-hypothesis, H0, is that the anomaly detection performance (in terms of precision or recall) when using aggregated neighborhood information is not

Table 3
PLE settings for further evaluation, based on Figs. 6 and 7.

Dataset	PLE_{opt}	$N()_{avg}$	PLE_{less}	$N()_{avg}$
GSB Humidity	4.0	5.3	6.2	2.5
GSB Temperature	4.0	5.3	6.2	2.5
Intel	3.4	5.8	6.2	2.7
Indoor WSN	4.6	4.1	6.2	1.2

better than the performance obtained by using only local information. Our alternative hypothesis, H1, is that the performance of the methods that include aggregated neighborhood information is better than when using only local information. Note that in the figures, we test also the case that a node has zero neighbors, because in the new method we did replace two input features, which might affect performance. We test the hypothesis for a significance of $\alpha = 0.1$ (or 10%) and $\alpha = 0.05$ (or 5%).

Fig. 8 shows that the precision is only significantly better with the Intel Lab dataset, while all other datasets show no improvement. Recall in Fig. 9, on the other hand, more often shows a significant improvement. In all cases, an improvement is visible with just few neighbors. Again the Intel Lab dataset shows a significant improvement in all cases. For the classifiers other than the RLS fusion classifier that include neighborhood information, and the ensembles, a similar pattern shows. That is, the methods show a significant increase in recall with the Intel Lab dataset only. The performance in the Intel Lab dataset can be well explained if we go back to Section 4.1, where we characterized the datasets in terms of average cross-correlation and in terms of spatial entropy. The Intel Lab dataset is the only dataset that has a relatively high cross-correlation value, and a low entropy. This leads us to conclude that the assumption (or requirement) of a similar mixture of diffusive processes is valid and, thus, that when neighborhood information correlates, the inclusion of aggregate neighborhood information in the prediction significantly improves the results.

4.3. Detection performance

With the above information, the detection performance of the other individual classifiers and ensembles thereof can be analyzed in more detail. Again we choose the mean and median neighborhood aggregation to replace the previous measurement and mean as input features in the fusion classifiers. In a real world deployment, often one cannot choose a perfect number of neighbors for each node. Therefore, we opt to evaluate two different PLE settings per dataset, to represent an optimal case in a dense network, and a less-than-optimal case in a sparse network. These choices are guided by Figs. 6, 7 and 9 and can be seen in Table 3.

In Table 4 we show the results of including neighborhood information with these settings, as percentage of improvement over the stand-alone methods. The absolute numbers can be found in Appendix B and the resulting F-measures in C.1. Here, too, we see that in general recall benefits from neighborhood information, and precision is similar or slightly reduced. The OS-ELM based classifier shows more extreme results due to its random initialization of input weights and biases, but does also benefit from neighborhood information. The results of the Intel Lab dataset show significant improvement for all classifiers. That is, statistical analysis in the line of Section 4.2.2 for these classifiers shows that, for all of them, there is a significant improvement in recall, but not necessarily in precision. Going back to the dataset characterization, and specifically to Table 2, we can observe that indeed the assumption of a similar mixture of diffusive processes is key to achieving good results. That is, the neighborhood correlation should be relatively high.

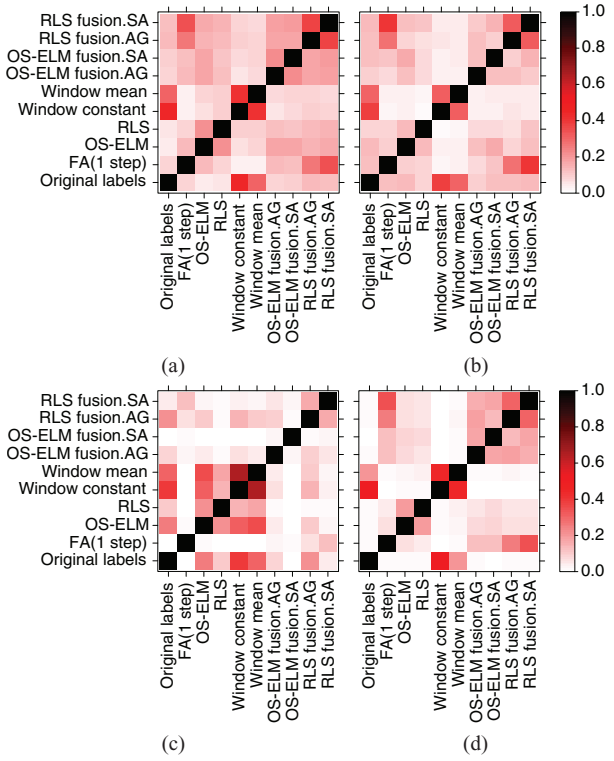

Fig. 10. Classifier Agreement, sum(and(*a, b*))/sum(or(*a, b*)), with PLE = 4.0 for the (a) GSB humidity, (b) GSB temperature, (c) Intel Lab, and (d) Indoor WSN datasets.

Choosing an optimum average neighborhood size also results in a better recall performance. The exception here is the GSB humidity dataset, which in Fig. 7a also showed a different trend (showing a peak around PLE = 6.2) and from Table 4 we see that this mainly concerns the recall. While this is most likely due to a difference in measured processes (where, for this dataset, the sensors relative humidity, soil moisture and watermark are better correlated with fewer near neighbors), these results are not significant.

Noteworthy is also the slight effect that neighborhood information has on the ensemble methods. The ensembles consist of multiple classifiers, of which only two (RLS and OS-ELM fusion) include neighborhood information. The other classifiers are the window constant, detecting if a constant anomaly occurs, and the 1-step-ahead function approximation classifier. From the dataset analysis, we know that 65 to 95% of the anomalies in the datasets is of the 'constant' anomaly type, and from previous investigations [54,56,57], we established that a simple rule-based classifier can detect these anomalies with very high accuracy. Therefore, we hypothesize that only few of the extra recalled anomalies are not of the constant type, and thus the ensembles do not benefit much more from the improved recall.

The agreement between classifiers is measured as the ratio of equality between two time series of logical values, *a* and *b*, divided by the total number of agreed detections possible between *a* and *b*, i.e., *sum*(*and*(*a, b*))/*sum*(*or*(*a, b*)). The logic values denote the detected or known anomalies in the time series. These ratios can be denoted in a confusion matrix and displayed similarly to the correlation maps in Section 4.1. For the sake of brevity, further information on classifier agreement is included in C.2. The resulting matrix is displayed graphically in Fig. 10. In this figure, the fusion classifiers that include neighborhood aggregate data are indicated with the postfix '.AG', and those that do not include neighborhood information (the stand-alone methods from our previous work [57]) with the postfix '.SA'.

Table 4
The impact of neighborhood information on recall is beneficially large. precision, however, suffers slightly from neighborhood information. Relative change in precision and recall, in percent.

x	GSB humidity PLE=4.0		GSB humidity PLE=6.2		GSB temperature PLE=4.0		GSB temperature PLE=6.2		Intellab PLE=3.4		Intellab PLE=6.2		Rwtb PLE=4.6		Rwtb PLE=6.2	
x	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re
RLS fusion	-0.17	9.78	0.40	17.06	2.50	31.66	-6.34	13.64	-2.70	213.58	-2.74	137.45	-4.90	26.93	-6.68	21.82
OS-ELM fusion	0.95	-5.67	-0.55	45.06	11.93	81.31	-12.88	26.51	-0.49	1557.88	-0.57	1172.76	6.83	86.88	-0.51	42.57
Ensemble (heuristic)	0.47	0.31	0.07	0.06	-1.94	0.24	-2.71	0.11	0.28	0.51	0.17	0.27	-1.42	0.78	-1.38	0.59
Fisher's method	0.29	0.19	0.05	0.18	0.03	0.16	-0.77	0.11	0.20	0.37	0.10	0.18	-0.13	0.58	-0.29	0.69
Ensemble (min)	0.64	0.37	-0.21	0.15	-2.10	0.21	-3.65	0.11	0.32	0.59	0.19	0.29	-1.51	1.11	-1.14	1.33
Ensemble (median)	1.15	-10.97	5.17	58.40	8.57	55.43	0.46	3.52	1.55	3709.42	-0.32	1267.51	0.98	21.22	-1.99	8.52

Fig. 10 shows that the constant anomaly agrees well with the original labels and with the window mean anomaly detector. Overall, the classifiers that include neighborhood information show moderate agreement between themselves and moderate to low agreement to the other classifiers, although a slight increase in agreement can be seen between the constant classifier and the RLS-fusion method that includes neighborhood information. The moderate to low agreement indicates that indeed the effect of neighborhood information on the median and the Fisher's method ensemble should be low, due to the nature of these ensembles. The median ensemble, however, does show a reasonable increase in performance in Table 4. Yet, previous work [57] showed that such an ensemble has high precision but very low recall, and thus improvements in recall are large in relative terms. The Intel Lab dataset here, too, stands out, as the fusion classifiers have lower agreement with the other classifiers than in the other datasets. That is, it seems the aggregate neighborhood data results in different anomalies being detected than those anomalies detected by the methods that use local data, which may explain the higher recall, as more different anomalies are detected. This again, shows that the assumption of a similar mixture of diffusive processes is important.

5. Discussion and conclusion

We have empirically shown that incorporating neighborhood information improves the anomaly detection, yet this is valid only in cases where the dataset is well-correlated and shows relatively low spatial entropy. These assumptions typically occur in the most common application of sensor networks, that of monitoring natural environments. In such a context, the above hypothesis is valid and there is significant detection performance benefit by using neighbor information. While this brings significant advantages in these cases (because the neighborhood information is correlated), other cases in which these assumptions do not hold will not benefit from aggregating neighborhood information (contrary to intuition). Thus, it is not always valuable to aggregate neighborhood information locally and it is often not valuable to aggregate among more than 5 neighbors since communication cost is high and the information gain saturates.

We explored this hypothesis in several steps. First, we showed that the above assumptions hold only to varying degree through the assessment of correlation in real-world data. Nevertheless, we showed that the mean and median aggregate operators are valid choices to reduce a dynamic neighborhood to a fixed measure that can be used in fusion methods. Next, we have evaluated the effect of neighborhood density (or communication range) on the quality of the data, by analyzing these effects on the RLS-fusion anomaly detector in simulation, with real-world datasets and topologies. This analysis showed that the amount of improvement mainly depends on the correlation within the dataset. This correlation is the result of the sensed processes, the type of sensors, the type of anomalies and the topology. Thus, the amount of improvement depends on the application. Nevertheless, the neighborhood information significantly contributed to the anomaly detection recall performance in the well-correlated dataset of the Intel Lab. The other datasets show a less, but significant, recall improvement with few neighbors. The precision performance, on the other hand, stayed equal or reduced moderately. Again, the exception is the Intel Lab dataset, which also benefited significantly.

Finally, the analysis of performance at a dense and a sparse network setting showed that adding neighborhood information to the RLS and OS-ELM fusion-based anomaly detectors shows a benefit in the recall. The ensemble methods, however, did not benefit greatly due to additional classifiers that did not make use of neighborhood data, and due to the constant anomaly dominating the anomalies, which is well detected by a simple rule.

The overall results show that, when a dataset stems from a similar mixture of diffusive processes (and thus is well-correlated), precision benefits, and a significant improvement in terms of recall can be established. However, one has to consider the target application (regarding sensors, anomalies and topology) to evaluate the need for local neighborhood information in online anomaly detection. In cases where a network is too sparse, or in cases where the environment under monitoring has no correlated diffusive processes, a local-only anomaly detection approach may be preferred to spare the limited resources available in an embedded context such as a WSN.

Future work may address the constraints on timely information sharing with a neighborhood: as the wireless communication is inherently unreliable, missing data may or may not affect the results significantly. Next to this, also slow diffusive processes may cause delays in correlated data. These constraints could be addressed by methods to automatically determine the delay in correlation in resource-limited platforms, or adopt a weighted aggregation approach over a larger time period to account for such correlation delays or differences. That is, the weight, a measure of relevance, could be determined with respect to time delays or correlation differences between nodes. Other questions that can be addressed are the use of aggregates or models as neighborhood information, instead of raw measurement data and the energy balance between more complex local processing and more decentralized local neighborhood communications.

Acknowledgment

This work was co-financed by the Province of Drenthe, the Municipality of Assen, the European Fund for Regional Development and the Ministry of Economic Affairs, Peaks in the Delta.

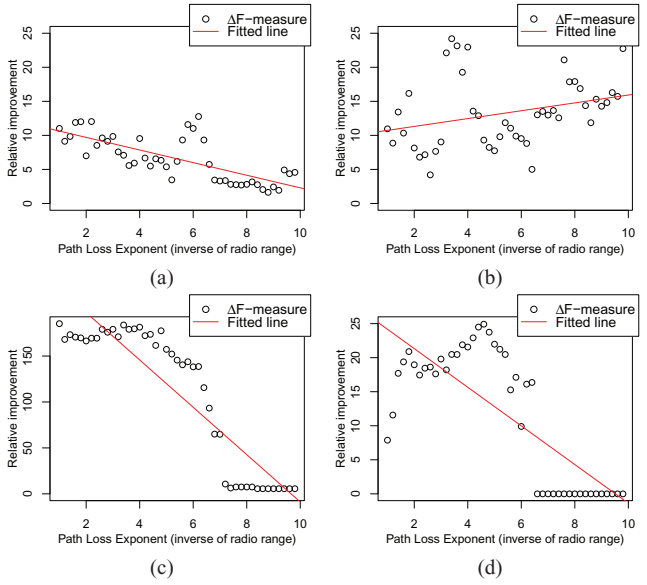

Fig. A.11. The PLE vs the relative change in F-Measure of the RLS-fusion classifier. The datasets are (a) GSB humidity, (b) GSB temperature, (c) Intel Lab (note the different y scale), and (d) Indoor WSN.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 665347.

Appendix A. PLE vs relative improvement

Fig. A.11 shows the F-measure improvement vs PLE setting. Fig. 7 was extracted from this figure and Fig. 6. Here, too, we see the optima differ per application scenario. The Intel Lab dataset also shows the highest relative improvement in F-measure of over 150%, which can be contributed to the dataset characteristic of being highly correlated. Moreover, from this figure we can see that each topology and application has its own optimal PLE setting, showing that transmission-power control may be beneficial in detection applications.

Appendix B. Absolute precision and recall

The absolute values of precision and recall, from which Table 4 is derived, are depicted in Table B.5. We see that in all cases, the offline baseline ensemble has the highest recall. Furthermore, we can see that in the case of the GSB humidity data and especially in the case of the Intel Lab data, the aggregate neighborhood information contributes significantly to the overall performance. For the fusion classifiers we see a clear benefit in recall for most datasets.

Appendix C. Detection performance

The following subsections were omitted from the main text, Section 4.3, for brevity.

C.1. Change in F-measure

Fig. 7 shows the relative change in F-measure for the RLS-fusion classifier. With the above settings, we further analyze the effect of neighborhood information on the OS-ELM-fusion based classifier, and the resulting effect on the ensemble classifiers. From this analysis, seen in Table C.6, we can see that not only for RLS but also for the OS-ELM based classifier including neighborhood information mostly has a positive benefit. However, the change for OS-ELM is more extreme. This is partly because the F-measure resulting from the anomaly detection without neighborhood information is low, specifically in the case of the Intel Lab dataset, so any change therein is relatively large. Another cause for the higher variability for the OS-ELM based detection is the random initialization of input weights and biases.

Furthermore, from Table C.6, we can see that the effect of the inclusion of neighborhood information on the ensembles is low. We hypothesize this has two reasons: First, the ensembles consists of a mix of classifiers, which include not only the RLS and OS-ELM fusion classifiers, but also the constant rule classifier and the 1-step-ahead function prediction classifier. The additional neighborhood information only affects the RLS and OS-ELM fusion classifiers and, therefore, the total effect on the ensembles is less. Second, the constant anomaly is the dominant anomaly, covering 83 to 95% of the anomalies in all datasets. Therefore, extra detections of anomalous samples is likely to be a constant anomaly which, thus, will not improve the ensemble score. This is further investigated in Section 4.3, where the classifier agreement is evaluated.

C.2. Classifier agreement

Finally, we evaluate the agreement between classifiers, to get a better understanding why the neighborhood information is of small influence on the ensemble classifiers. The behavior of the median ensemble and the Fisher's method ensemble is that when multiple classifiers agree on a sample being anomalous, the more likely it is to be anomalous. Thus, if more classifiers detect the same sample as anomalous, the better the performance of these ensembles. The minimum p-value and heuristic ensemble operate differently. The former is not influenced by multiple classifiers that judge similarly, but only returns the minimum of the p-values within the ensembled classifiers. Thus, this ensemble would benefit from more confident classifiers. Such an approach should improve recall, but the precision may suffer. The heuristic ensemble combines the constant rule and the RLS-fusion classifier. When a constant is detected by the constant rule, the RLS-fusion classification is ignored. Therefore, when the latter detects a constant anomaly, the performance of the heuristic ensemble is not improved.

The agreement between classifiers is measured as the ratio of equality between two time series of logical values, a and b, divided by the total number of agreed detections possible between a and b, i.e., $sum(and(a, b))/sum(or(a, b))$. The logic values denote the detected or known anomalies in the time series. These ratios can be denoted in a confusion matrix and displayed similarly to the correlation maps in Section 4.1.

The resulting matrix is displayed graphically in Fig. 10. In this figure, the fusion classifiers that do not include neighborhood information are indicated by the postfix '.SA', signifying the approach from our previous work [57]. The figure shows that the constant anomaly agrees well with the original labels and the window mean anomaly detector. The other classifiers do not agree much with the constant classifier, although a slight increase in agreement can be seen between it and the RLS-fusion method that includes neighborhood information. Furthermore, the figure shows that Function Approximation classifier agrees reasonably with the fusion classifiers, which is the result of the FA prediction being included in the input of the fusion classifiers. The fusion methods agree more among each other. Their agreement, however, becomes lower when neighborhood information is included. That would mean that the

Table B.5
Absolute precision/recall numbers. The baseline LLSE and ELM are the non-iterative variants of RLS and OS-ELM respectively. The postfix (d) denotes the inclusion of a day period. The postfix '.SA' indicates the results of the stand-alone methods described in previous work [57], while the postfix '.AG' indicates the current results, including the neighborhoog aggregates.

classifier \ dataset	GSB Humidity PLE = 4.0		GSB Humidity PLE = 6.2		GSB Temperature PLE = 4.0		GSB Temperature PLE = 6.2		Intel Lab PLE = 3.4		Intel Lab PLE = 6.2		Indoor WSN PLE = 4.6		Indoor WSN PLE = 6.2	
metric	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re	pr	re
FA(1 step)	66.69	5.23	66.53	5.23	53.49	9.22	53.49	9.22	89.71	0.59	89.71	0.59	38.57	7.97	38.57	7.97
OS-ELM	11.23	10.50	13.32	13.77	77.83	24.13	47.14	19.24	48.64	58.76	49.41	69.73	15.38	13.35	16.06	12.57
RLS	16.68	16.16	16.68	16.16	80.21	20.72	80.21	20.72	45.32	28.26	45.32	28.26	13.48	10.79	13.48	10.79
Window constant	41.35	83.56	41.35	83.56	**97.81**	88.72	**97.81**	88.72	47.05	95.30	47.05	95.30	**54.14**	63.48	**54.14**	63.48
OS-ELM fusion.AG	**92.59**	15.30	86.39	11.27	85.60	18.28	55.65	5.25	98.64	5.15	98.46	3.58	46.32	5.72	46.82	3.92
OS-ELM fusion.SA	91.72	16.23	86.87	7.77	76.48	10.09	63.88	4.15	99.14	0.31	**99.02**	0.28	43.36	3.06	47.06	2.75
RLS fusion.AG	90.12	25.15	90.63	26.82	61.61	18.05	56.30	15.58	95.99	14.87	95.95	11.26	42.68	8.72	41.88	8.38
RLS fusion.SA	90.26	22.91	90.27	22.91	60.11	13.71	60.11	13.71	98.65	4.74	98.65	4.74	44.87	6.87	44.88	6.87
Fisher's method.AG	41.91	85.78	41.82	85.66	93.35	91.49	92.78	91.45	49.23	95.30	49.18	95.12	52.03	65.60	52.02	65.59
Fisher's method.SA	41.79	85.62	41.80	85.51	93.33	91.34	93.50	91.36	49.13	94.95	49.13	94.95	52.09	65.22	52.16	65.14
Ensemble (heuristic).AG	42.32	86.88	42.14	86.66	88.03	92.37	87.34	92.25	47.25	95.98	47.20	95.75	52.05	67.12	52.07	66.99
Ensemble (heuristic).SA	42.11	86.61	42.11	86.61	89.76	92.15	89.76	92.15	47.12	95.49	47.12	95.49	52.80	66.60	52.80	66.60
Ensemble (min).AG	42.64	87.56	42.34	87.29	84.41	92.80	83.50	92.76	47.30	96.17	47.24	95.90	50.12	68.24	50.42	68.39
Ensemble (min).SA	42.37	87.24	42.43	87.16	86.22	92.61	86.66	92.66	47.15	95.62	47.15	95.62	50.89	67.49	51.00	67.49
Ensemble (median).AG	91.09	8.12	**91.06**	8.08	90.23	8.10	77.98	3.53	**99.75**	4.34	98.40	3.17	47.58	2.91	48.78	2.42
Ensemble (median).SA	90.05	9.12	86.58	5.10	83.10	5.21	77.62	3.41	98.24	0.11	98.71	0.23	47.12	2.40	49.78	2.23
Baseline rule	42.12	87.19	42.12	87.19	90.86	93.31	90.86	93.31	44.62	96.49	44.62	96.49	52.46	66.69	52.46	66.69
Baseline LLSE	66.52	9.64	66.52	9.64	45.28	16.06	45.28	16.06	80.80	8.12	80.80	8.12	32.62	11.41	32.62	11.41
Baseline LLSE (d)	67.72	10.84	67.72	10.84	42.73	16.04	42.73	16.04	75.07	7.78	75.07	7.78	31.19	11.36	31.19	11.36
Baseline ELM	62.86	9.55	63.40	9.39	41.45	17.86	42.75	17.93	79.89	8.12	80.35	7.98	33.84	11.74	32.61	11.31
Baseline ELM (d)	59.80	9.58	63.66	10.02	40.13	17.23	41.50	17.13	73.66	7.37	74.64	7.23	32.77	11.74	32.03	11.60
Baseline ensemble	44.16	**89.36**	44.17	**89.25**	74.85	**94.53**	75.86	**94.55**	44.96	**97.39**	44.95	**97.37**	47.89	**70.52**	47.82	**70.52**

Table C.6
The fusion classifiers are positively affected by neighborhood information, but the end result on most ensembles in negligible. The table shows relative change in F-Measure, in percent.

classifier \ dataset	GSB Humidity PLE=4.0	GSB Humidity PLE=6.2	GSB Temperature PLE=4.0	GSB Temperature PLE=6.2	Intel Lab PLE=3.4	Intellab PLE=6.2	Indoor WSN PLE=4.6	Rwtb PLE=6.2
metric	fm	fm	fm	fm	fm	fm	fm	fm
RLS fusion	7.61	13.24	25.08	9.32	184.51	122.73	21.57	17.11
OS-ELM fusion	-4.75	39.83	69.08	23.10	1482.04	1131.55	77.98	39.28
Ensemble (heuristic)	0.42	0.07	-0.88	-1.33	0.35	0.21	-0.46	-0.53
Fisher's method	0.25	0.09	0.10	-0.34	0.26	0.14	0.19	0.14
Ensemble (min)	0.54	-0.09	-1.01	-1.88	0.41	0.22	-0.40	-0.09
Ensemble (median)	-9.96	54.06	51.57	3.21	3559.71	1227.27	19.90	8.20

OS-ELM and RLS fusion detect different (types) of anomalies using neighborhood information.

Overall, the classifiers that include neighborhood information show moderate agreement between themselves and moderate to low agreement to the other classifiers. The effect of neighborhood information on the median and the Fisher's method ensemble should, therefore, be low. The median ensemble, however, does show a reasonable increase in performance in Table C.6 and 4. But, previous work showed that such an ensemble has high precision but very low recall, and thus improvements in recall are large in relative terms. The minimum p-value ensemble would benefit only from higher confidence in the detection of a single classifier, which cannot be tested by classifier agreement. The heuristic ensemble should benefit from neighborhood information in the RLS-fusion classifier, if this classifier has little agreement with the constant classifier. However, from Fig. 10 we see that the RLS-fusion has more agreement with the constant classifier when including neighborhood information, compared to the stand-alone method without neighborhood information. Therefore, their detections may overlap, and the ensemble may not benefit from the neighborhood information. This is in agreement with the earlier evaluation of F-measure, prediction and recall changes in previous sections.

References

[1] A. Whitmore, A. Agarwal, L. Da Xu, The internet of thingsa survey of topics and trends, Inf. Syst. Front. 17 (2) (2015) 261–274, doi:10.1007/s10796-014-9489-2.
[2] G. Fortino, A. Guerrieri, G.M.P. O'Hare, A.G. Ruzzelli, A flexible building management framework based on wireless sensor and actuator networks, J. Netw. Comput. Appl. 35 (6) (2012) 1934–1952, doi:10.1016/j.jnca.2012.07.016.
[3] M.F. Othman, K. Shazali, Wireless sensor network applications: a study in environment monitoring system, Proc. Eng. 41 (2012) 1204–1210 International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), doi:10.1016/j.proeng.2012.07.302.
[4] O. Demigha, W.-K. Hidouci, T. Ahmed, On energy efficiency in collaborative target tracking in wireless sensor network: a review, Commun. Surv. Tut. IEEE 15 (3) (2013) 1210–1222, doi:10.1109/SURV.2012.042512.00030.
[5] S. Sivavakeesar, G. Pavlou, C. Bohoris, A. Liotta, Effective management through prediction-based clustering approach in the next-generation ad hoc networks, in: Communications, 2004 IEEE International Conference on, vol. 7, 2004, pp. 4326–4330, doi:10.1109/ICC.2004.1313364.
[6] V. Menkovski, G. Exarchakos, A. Liotta, Machine learning approach for quality of experience aware networks, in: Intelligent Networking and Collaborative Systems (INCOS), 2010 2nd International Conference on, 2010, pp. 461–466, doi:10.1109/INCOS.2010.86.
[7] N.N. Qadri, A. Liotta, Pervasive Computing: Innovations in Intelligent Multimedia and Applications, Springer, London, pp. 433–453. 10.1007/978-1-84882-599-4_19
[8] G. Fortino, M. Bal, W. Li, W. Shen, Collaborative wireless sensor networks: architectures, algorithms and applications, Inf. Fusion 22 (2015) 1–2, doi:10.1016/j.inffus.2014.03.004.
[9] A. Liotta, D. Geelen, G. van Kempen, F. van Hoogstraten, A survey on networks for smartmetering systems, Int. J. Pervasive Comput.Commun. 8 (1) (2012) 23–52, doi:10.1108/17427371211221072.
[10] A. Liotta, The cognitive net is coming, IEEE Spect. 50 (8) (2013) 26–31.
[11] A. Liotta, G. Pavlou, G. Knight, Exploiting agent mobility for large-scale network monitoring, IEEE Netw. 16 (3) (2002) 7–15, doi:10.1109/MNET.2002.1002994.
[12] G. Liu, R. Tan, R. Zhou, G. Xing, W.-Z. Song, J.M. Lees, Volcanic earthquake timing using wireless sensor networks, in: Proceedings of the 12th International Conference on Information Processing in Sensor Networks, ACM, 2013, pp. 91–102.
[13] G. Fortino, R. Giannantonio, R. Gravina, P. Kuryloski, R. Jafari, Enabling effective programming and flexible management of efficient body sensor network applications, IEEE Trans. Human Machine Syst. 43 (1) (2013) 115–133, doi:10.1109/TSMCC.2012.2215852.
[14] A. Alzghoul, M. Löfstrand, B. Backe, Data stream forecasting for system fault prediction, Comput. Indus. Eng. 62 (4) (2012) 972–978, doi:10.1016/j.cie.2011.12.023.
[15] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: a survey, ACM Comput. Surv. (CSUR) 41 (3) (2007/2009) 1–58.
[16] C. Phua, V. Lee, K. Smith, R. Gayler, A comprehensive survey of data mining-based fraud detection research, in: 2010 International Conference on Intelligent Computation Technology and Automation (ICICTA), 2010, pp. 1–14. abs/1009.6119
[17] S. Kao, A. Ganguly, K. Steinhaeuser, Motivating complex dependence structures in data mining: a case study with anomaly detection in climate, in: 2009 IEEE International Conference on Data Mining Workshops, IEEE, 2009, pp. 223–230.
[18] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, M. Rajarajan, A survey of intrusion detection techniques in cloud, J. Netw. Comput. Appl. 36 (1) (2013) 42–57.
[19] L. Zeng, L. Li, L. Duan, K. Lu, Z. Shi, M. Wang, W. Wu, P. Luo, Distributed data mining: a survey, Inf. Technol. Manage. 13 (4) (2012) 403–409.
[20] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, L. Mo, W. Dong, Z. Yang, M. Xi, J. Zhao, et al., Does wireless sensor network scale? A measurement study on greenorbs, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 873–881.
[21] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez, E. Vazquez, Anomaly-based network intrusion detection: techniques, systems and challenges, Comput.Secur. 28 (1-2) (2009) 18–28.
[22] C. Wang, K. Viswanathan, L. Choudur, V. Talwar, W. Satterfield, K. Schwan, Statistical techniques for online anomaly detection in data centers, in: Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on, IEEE, 2011, pp. 385–392.
[23] S. Budalakoti, A. Srivastava, M. Otey, Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety, IEEE Trans. Systems Man Cybernetics Part C Appl. Rev. 39 (1) (2009) 101–113.
[24] D.-I. Curiac, C. Volosencu, Ensemble based sensing anomaly detection in wireless sensor networks, Expert Systems with Applications 39 (10) (2012) 9087–9096.
[25] F. Serdio, E. Lughofer, K. Pichler, T. Buchegger, M. Pichler, H. Efendic, Fault detection in multi-sensor networks based on multivariate time-series models and orthogonal transformations, Inf. Fusion 20 (2014) 272–291, doi:10.1016/j.inffus.2014.03.006.
[26] F. Gaillard, E. Autret, V. Thierry, P. Galaup, C. Coatanoan, T. Loubrieu, Quality control of large argo datasets, J. Atm. Ocean. Technol. 26 (2) (2009) 337–351.
[27] K. Langendoen, A. Baggio, O. Visser, Murphy loves potatoes: experiences from a pilot sensor network deployment in precision agriculture, in: Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International, IEEE, 2006, pp. 8–pp.
[28] M. Abu Alsheikh, S. Lin, D. Niyato, H.-P. Tan, Machine learning in wireless sensor networks: algorithms, strategies, and applications, Commun. Surv. Tut. IEEE 16 (4) (2014) 1996–2018, doi:10.1109/COMST.2014.2320099.
[29] T.A. Nguyen, D. Bucur, M. Aiello, K. Tei, Applying time series analysis and neighbourhood voting in a decentralised approach for fault detection and classification in wsns, in: Proceedings of The 4th International Symposium on Information and Communication Technology, ACM, 2013, pp. 234–241.
[30] M. Chang, A. Terzis, P. Bonnet, Mote-based online anomaly detection using echo state networks, in: Distributed Computing in Sensor Systems, Springer, 2009, pp. 72–86.
[31] G. Fortino, S. Galzarano, R. Gravina, W. Li, A framework for collaborative computing and multi-sensor data fusion in body sensor networks, Inf. Fusion 22 (2015) 50–70, doi:10.1016/j.inffus.2014.03.005.

[32] H. Luo, J. Luo, Y. Liu, Energy efficient routing with adaptive data fusion in sensor networks, in: Proceedings of the 2005 joint workshop on Foundations of mobile computing, ACM, 2005, pp. 80–88.
[33] J. Cabrera, C. Gutiérrez, R. Mehra, Ensemble methods for anomaly detection and distributed intrusion detection in mobile ad-hoc networks, Inf. Fusion 9 (1) (2008) 96–119.
[34] H. Kumarage, I. Khalil, Z. Tari, A. Zomaya, Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling, J. Parallel Distrib. Comput. 73 (6) (2013) 790–806.
[35] S. Rajasegarar, C. Leckie, M. Palaniswami, J. Bezdek, Distributed anomaly detection in wireless sensor networks, in: Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on, IEEE, 2006, pp. 1–5.
[36] D.P. Spanos, R. Olfati-Saber, R.M. Murray, Distributed sensor fusion using dynamic consensus, IFAC World Congress, Prague Czech Republic, 2005.
[37] T.C. Aysal, M.J. Coates, M.G. Rabbat, Distributed average consensus with dithered quantization, IEEE Trans. Signal Process. 56 (10) (2008) 4905–4918.
[38] H.J. LeBlanc, H. Zhang, S. Sundaram, X. Koutsoukos, Consensus of multi-agent networks in the presence of adversaries using only local information, in: Proceedings of the 1st international conference on High Confidence Networked Systems, ACM, 2012, pp. 1–10.
[39] J.W. Branch, C. Giannella, B. Szymanski, R. Wolff, H. Kargupta, In-network outlier detection in wireless sensor networks, Knowl.Info.Syst. 34 (1) (2013) 23–54.
[40] K. Flouri, B. Beferull-Lozano, P. Tsakalides, Distributed consensus algorithms for svm training in wireless sensor networks, in: Signal Processing Conference, 2008 16th European, 2008, pp. 1–5.
[41] S. Barbarossa, G. Scutari, A. Swami, Achieving consensus in self-organizing wireless sensor networks: the impact of network topology on energy consumption, in: Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on, vol. 2, IEEE, 2007, pp. II–841.
[42] E. Schubert, A. Zimek, H.-P. Kriegel, Local outlier detection reconsidered: a generalized view on locality with applications to spatial, video, and network outlier detection, Data Mining Knowl. Disc. 28 (1) (2014) 190–237.
[43] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander, Lof: identifying density-based local outliers, in: ACM Sigmod Record, vol. 29, ACM, 2000, pp. 93–104.
[44] S. Chawla, P. Sun, Slom: a new measure for local spatial outliers, Knowl. Inf. Syst. 9 (4) (2006) 412–429.
[45] F. Chen, C.-T. Lu, A.P. Boedihardjo, Gls-sod: a generalized local statistical approach for spatial outlier detection, in: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2010, pp. 1069–1078.
[46] D. McDonald, S. Sanchez, S. Madria, F. Ercal, A survey of methods for finding outliers in wireless sensor networks, J. Netw. Sys. Manage. (2013) 1–20.
[47] N. Chitradevi, V. Palanisamy, K. Baskaran, K. Swathithya, Efficient density based techniques for anomalous data detection in wireless sensor networks, J. Appl. Sci. Eng. 16 (2) (2013) 211223.
[48] L. Xu, Y.-R. Yeh, Y.-J. Lee, J. Li, A hierarchical framework using approximated local outlier factor for efficient anomaly detection, Proc. Comput. Sci. 19 (2013) 1174–1181.
[49] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, P. Deng, Localized outlying and boundary data detection in sensor networks, IEEE Trans.Knowl. Data Eng. 19 (8) (2007) 1145–1157.
[50] L. Fang, S. Dobson, Data collection with in-network fault detection based on spatial correlation, in: Cloud and Autonomic Computing (ICCAC), 2014 International Conference on, IEEE, 2014, pp. 56–65.
[51] P. Wang, T. Wang, Adaptive routing for sensor networks using reinforcement learning, in: Computer and Information Technology, 2006. CIT'06. The Sixth IEEE International Conference on, IEEE, 2006.219–219
[52] A.M. Zungeru, L.-M. Ang, K.P. Seng, Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison, J. Netw. Comput.r Appl. 35 (5) (2012) 1508–1536 Service Delivery Management in Broadband Networks, doi:10.1016/j.jnca.2012.03.004.
[53] Y. Zhang, N. Meratnia, P.J. Havinga, Distributed online outlier detection in wireless sensor networks using ellipsoidal support vector machine, Ad hoc Netw. 11 (3) (2013) 1062–1074.
[54] H.H.W.J. Bosman, A. Liotta, G. Iacca, H.J. Wortche, Anomaly detection in sensor systems using lightweight machine learning, in: Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on, IEEE, 2013a, pp. 7–13.
[55] H.H.W.J. Bosman, A. Liotta, G. Iacca, H.J. Wortche, Online extreme learning on fixed-point sensor networks, in: Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on, IEEE, 2013b, pp. 319–326.
[56] H.H.W.J. Bosman, G. Iacca, H.J. Wörtche, A. Liotta, Online fusion of incremental learning for wireless sensor networks, in: Data Mining Workshop (ICDMW), 2014 IEEE International Conference on, 2014, pp. 525–532.
[57] H.H. Bosman, G. Iacca, A. Tejada, H.J. Wörtche, A. Liotta, Ensembles of incremental learners to detect anomalies in ad hoc sensor networks, Ad Hoc Netw. 35 (2015) 14–36 Special Issue on Big Data Inspired Data Sensing, Processing and Networking Technologies, doi:10.1016/j.adhoc.2015.07.013.
[58] L.G. Birta, G. Arbez, Modelling and Simulation, Simulation Foundations, Methods and Applications, Springer, 2013, doi:10.1007/978-1-4471-2783-3.
[59] S. Basu, N. Kumar, Modelling and Simulation of Diffusive Processes, Simulation Foundations, Methods and Applications, Springer International Publishing Switzerland, 2014, doi:10.1007/978-3-319-05657-9.
[60] F. Cauteruccio, G. Fortino, A. Guerrieri, G. Terracina, Internet and Distributed Computing Systems: 7th International Conference, IDCS 2014, Calabria, Italy, September 22-24, 2014. Proceedings, Springer International Publishing, Cham, pp. 383–395. doi:10.1007/978-3-319-11692-1_33.
[61] K. Romer, B.-C. Renner, Aggregating sensor data from overlapping multi-hop network neighborhoods: Push or pull? in: Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on, IEEE, 2008, pp. 107–110.
[62] J. Lee Rodgers, W.A. Nicewander, Thirteen ways to look at the correlation coefficient, Am. Statist. 42 (1) (1988) 59–66.
[63] N.C. Silver, W.P. Dunlap, Averaging correlation coefficients: should fisher's z transformation be used? J. Appl. Psychol. 72 (1) (1987) 146.
[64] M. Batty, R. Morphet, P. Masucci, K. Stanilov, Entropy, complexity, and spatial information, J. Geograph. Syst. 16 (4) (2014) 363–385, doi:10.1007/s10109-014-0202-2.
[65] M. De Berg, M. Van Kreveld, M. Overmars, O.C. Schwarzkopf, Computational-Geometry, Springer Berlin Heidelberg, 2000.
[66] H. Lee, A. Cerpa, P. Levis, Improving wireless simulation through noise modeling, in: Information Processing in Sensor Networks, 2007. IPSN 2007. 6th International Symposium on, 2007, pp. 21–30.
[67] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: a new learning scheme of feedforward neural networks, in: Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on, vol. 2, IEEE, 2004, pp. 985–990.
[68] Y.-H. Pao, G.-H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[69] E. Fuchs, T. Gruber, J. Nitschke, B. Sick, Online segmentation of time series based on polynomial least-squares approximations, IEEE Trans.Pattern Anal. Mach. Intell. 32 (12) (2010) 2232–2245.
[70] P. Levis, N. Lee, M. Welsh, D. Culler, Tossim: accurate and scalable simulation of entire tinyos applications, in: Proceedings of the 1st international conference on Embedded networked sensor systems, ACM, 2003, pp. 126–137.
[71] A. Sharma, L. Golubchik, R. Govindan, On the prevalence of sensor faults in real-world deployments, in: Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on, IEEE, 2007, pp. 213–222.
[72] D.M. Powers, Evaluation: from precision, recall and f-factor to roc, informedness, markedness & correlation, Evaluation (2007).
[73] F.J. Massey Jr, The Kolmogorov–Smirnov test for goodness of fit, J. Am.Statist. Assoc. 46 (253) (1951) 68–78.