Omnibus Outlier Detection in Sensor Networks using Windowed Locality Sensitive Hashing

Nikos Giatrakos[a,b,*], Antonios Deligiannakis[a,b], Minos Garofalakis[a,b], Yannis Kotidis[c]

[a] *ATHENA Research and Innovation Centre, Artemidos 6 & Epidavrou, GR-15125 Athens, Greece*
[b] *Department of Electrical and Computer Engineering, Technical University of Crete, University Campus., GR-73100 Chania, Greece*
[c] *Department of Informatics, Athens University of Economics and Business, 76 Patission St., GR-10434 Athens, Greece*

Abstract

Wireless Sensor Networks (WSNs) have become an integral part of cutting edge technological paradigms such as the Internet-of-Things (IoT) which incorporates a variety of smart application scenarios. WSNs include tiny sensors (motes), with constrained hardware capabilities and limited power supply that can collaboratively function in an unsupervised manner for a long period of time. Their purpose is to continuously monitor quantities of interest and provide answers to application queries. Sensor data streams are inherently spatiotemporal in nature both because mote measurements form multidimensional time series and due to the spatial reference on the data based on the realm sensed by a mote. Motes are designed to be inexpensive, and thus sensory hardware is prone to temporary or permanent failures yielding faulty measurements. Such measurements may unpredictably forge a query answer, while truthful but abnormal mote samples may indicate undergoing phenomena. Therefore, outlier detection in sensor networks is of utmost importance.

With limited power supply and communication being by far the main culprit in energy drain, outlier detection techniques in WSNs should achieve appropriate balance between reducing communication and providing real-time, continuously updated outlier reports. Prior works employ probabilistic or best effort approaches to accomplish the task, which either unpredictably compromise outlier detection accuracy or fail to explicitly tune the amount of communicated data. In this work, we introduce an omnibus outlier detection solution over spatiotemporally referenced sensor data that is capable of: (a) directly trading communication reduction for outlier detection quality with predictable accuracy guarantees, (b) accommodating both uni- and multi-dimensional outlier definitions, (c) operating under various streaming window models and (d) incorporating a wide variety of similarity measures to judge outliers.

Keywords: Sensor Network, Outlier, Locality Sensitive Hashing, Streaming Window Model, Similarity Estimation

1. Introduction

Emerging technological paradigms such as the Internet-of-Things (IoT) promise to flood the world with computing devices that will effectively collaborate to transform platforms and environments into sensitive, responsive and smart systems. Cutting edge technology related to smart grids, smart cities, intelligent transportation, remote patient health monitoring are infrastructure systems the common vision of which is associated with IoT. In IoT platforms a wide area, physical infrastructure is configured to effectively perform intelligent monitoring and information management via the usage of networked devices. Devices are interconnected to transmit useful measurement information and control instructions via distributed sensor networks. Therefore, wireless sensor networks recently find themselves needed in a wide variety of so-called "smart" application fields. Wireless sensor networks consist of tiny sensor nodes (motes) equipped with limited memory and CPU capacity, wireless communication (of few tens of meters range) infrastructure and constrained power supply. Motes are placed in environments of interest where they are required to

*Corresponding author. Tel.: (+30) 28210 37265; Fax: (+30) 28210 37542
Email addresses: ngiatrakos@imis.athena-innovation.gr (Nikos Giatrakos), adeli@softnet.tuc.gr (Antonios Deligiannakis), minos@imis.athena-innovation.gr (Minos Garofalakis), kotidis@aueb.gr (Yannis Kotidis)

self-organize and function in an unsupervised manner for a long period of time. Together with the aforementioned elements, sensory devices placed on motes are capable of sensing their surrounding and provide information about a variety of features such as temperature, humidity, soil texture, noise levels, velocity, acceleration of pinpointed moving objects etc. Acquired measurements of attributes are taking place in continuous data processing procedures [1] (e.g., aggregate query answering) during which they are locally processed and are subsequently communicated to a distant base station in a multi-hop fashion.

Sensor data are inherently spatiotemporal in nature. Motes periodically collect measurements describing their surrounding forming respective, multidimensional time series. Moreover, as is the case with IoT platforms and other large scale application fields, sensor networks monitor a wide area with each device being appointed to sensing its own realm, thus attributing an additional spatial dimension, as a reference to the collected temporal data. Therefore, analyzing distributed datasets stemming from various motes mainly involves discovering patterns and structures by diving into spatial and temporal correlations of neighboring or distant sensors' time variant readings.

In principle, sensor nodes are designed to be inexpensive and, as such, their sensing particles often malfunction or occasionally acquire spurious values causing a distortion on the monitored features [2, 3, 4, 5]. Similar noisy measurements can also be obtained due to environmental interference [6, 4]. Motes with extraordinary samples stemming from such hardware failures should be detected and get isolated since they may forge the outcome of query processing [2]. On the other hand, abnormal values in acquired samples may represent interesting undergoing phenomena within the sensor network setting, such as a fire burst or a flood.

A tremendous effort has been devoted to accomplishing the challenging task of outlier detection in sensor network environments [7, 2, 3, 4, 5]. The difficulty in developing outlier detection techniques in the resource constrained, sensor setting mainly stems from the existence of conflicting requirements. Pinpointed outliers need to be delivered in a *continuous* fashion, as (due to the rapidly changing data distributions of mote produced streams) the results may quickly become outdated. Moreover, motes cannot individually determine if they sample extraordinary measurements and need to examine their similarity with neighboring motes to *support* their obtained measurements, as spatial correlations exist for motes monitoring the same surrounding [2, 3, 4, 5]. While the demand for continuous outlier reports through correlation discovery entails communication of sensor measurements, data transmission and reception are the main factor of energy drain [1] for the battery powered sensors. Should sensors deplete their residual energy, the whole networks' connectivity is lost thus decreasing its lifetime. Therefore, outlier detection techniques in sensor networks should achieve appropriate balance between reducing communication and providing real-time, continuously updated outlier reports.

Prior works [7] attempt to incorporate these constraints and reduce communication following, for instance, probabilistic or best effort approaches [8, 2]. Nevertheless, all these approaches attempt to place certain criteria under which communication between motes (directly or through a parent node) will take place and apply some kind of heuristics (e.g., election of motes producing the most representative samples) to probe only a subset of the sensor nodes. However, *none of them provides explicit guarantees, neither regarding the amount of data that they require to be communicated in the sensor network nor with respect to the accuracy of the extracted outlier sensors*.

Prior Work. The TACO framework [4, 5] was the first to introduce an outlier detection technique that satisfies both the aforementioned criteria. A plausible observation made in TACO as well as in most of related works [7, 9] is that outlier detection should be conducted distributively and *in-network*, as the central collection of data in a basestation is infeasible due to the cumulative effect in the amount of transmitted data in a multi-hop fashion. TACO formulates an in-network, continuous outlier identification procedure which utilizes a Locality Sensitive Hashing (LSH) [10] based data compression scheme to reduce the amount of transmitted data and simultaneously provide the means to predict the accuracy of the similarity tests between motes upon using those compressed representations. As a result, the future network operation and the outlier accuracy can be forecasted to the major extend. In particular, TACO first utilizes LSH in order to encode the latest ω measurements collected by each sensor node as a bitmap of $d \ll \omega$ bits. This encoding is performed locally at each node. The encoding that is utilized trades accuracy for bandwidth, by simply varying the desired level of dimensionality reduction and provides tunable accuracy guarantees based on the d parameter mentioned above. Assuming a clustered network organization [11, 12, 13, 14], motes communicate their bitmaps to their clusterhead, which can estimate the similarity amongst the latest values of any pair of sensors in its cluster by comparing their bitmaps. Based on the performed similarity tests, and a desired *minimum support* specified by the posed query, each clusterhead generates a list of *potential* outlier nodes. At a second (inter-cluster) phase of the

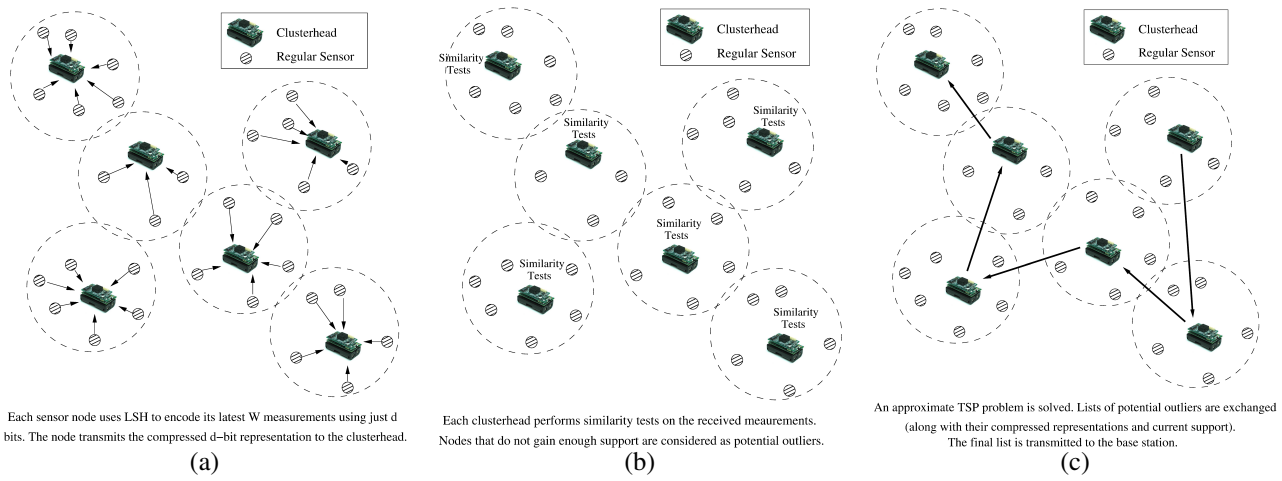


Figure 1: Main Stages of our Outlier Detection Process

algorithm, this list is then communicated among the clusterheads, in order to allow potential outliers to gain support from measurements of nodes that lie within other clusters. TACO can incorporate a variety of metrics for judging mote similarity that may be useful in a wide range of applications. The whole process is sketched in Figure 1.

Our Contributions. Although the framework in [4, 5] manages to predictably balance conflicting constraints in a tunable manner, it also consolidates important limitations. Initially, it is capable of detecting outlier sensor nodes taking into account only one dimension - monitored attribute. Nonetheless, weird samples' determination based on multidimensional outlier definitions can find itself quite needed and in fact the inability of certain techniques to operate in multiple dimensions has been pointed out as an important drawback in comparative analyses [7]. From an application viewpoint, a handy and plausible observation is that in most practical scenarios, a distinction between dirty readings and interesting phenomena can be achieved by reckoning that dirty readings usually appear on individual sensory elements. For instance, a rain drop can cause the humidity sensor of the popular MICA mote to report spikes [2], while dusty temperature sensories are evidently uncapable of observing truthful environmental conditions. In contrast to the case of dirty mote readings, unusual samples depicting interesting phenomena more often than not affect the whole bunch of monitored quantities. As an example, a fire burst near a mote will cause its temperature and radiance values to steeply increase, while humidity values will be downgraded. Additionally, in an industrial application, all motes placed on a malfunctioning production machine will experience significant vibrations, high noise as well as heat levels.

A second limitation of the framework in [4, 5] is that it operates on a per tumble fashion, where tumbles [15] are defined as disjoint windows of observations obtained by each mote. Many applications however need to base their decisions on long term observations [16] and hence, dictate the adoption of large windows of acquired samples. Utilizing a tumble in such cases would cause delays and it would hinder the continuous delivery of outlier results with high frequency, since for each tumble motes need to first fill in an, each time, emptied window. Evidently, in such cases a sliding window (i.e. a window where the most aged observation expires and the latest one is appended to it) appears the most suitable choice that [4, 5] fail to consolidate.

Concentrating on the previously discussed aspects, in this work we overcome this pair of limitations. The integration of multidimensional outlier detection and sliding window operation by our approaches leads to superior results since by tackling with each limitation separately, we not only achieve to override it, but also attribute extra features that are missing from any related framework. In particular, by enabling multidimensional outlier detection and sliding window operation we end up with four modes of operation during the outlier detection process. Namely, (a) unidimensional outlier definition and tumble operation as in prior work [4, 5], (b) multidimensional outlier definition and tumble operation, (c) unidimensional outlier definition and sliding window operation and finally (d) multidimensional outlier definition and sliding window operation. From the above operational modes, (b), (c), (d) are introduced in this work. Additionally, we show that our techniques preserve the ability to i) trade bandwidth for accuracy in a

straightforward manner and ii) employ a variety of commonly used similarity measures for judging outliers and their generalizations to multidimensional cases. Consequently, the suite of algorithms we propose in this work comprises an omnibus outlier detection solution for wireless sensor networks with tunable quality guarantees.

The proposed approaches are straightforwardly related to time series data compression under various temporal (window) models, while leveraging the spatial dimension of the data both at the architectural level via the employed clustered sensor network scheme, as well as during the intra- and inter-cluster processing phases. Based on these, our techniques provide communication efficient algorithms for large-scale (for instance, in IoT), distributed implementations of perhaps the most interesting data mining task in the sensor network setting, namely real-time outlier detection. It is important to note that our techniques can easily be adapted to serve other analytics tasks in resource constrained environments that require distributed, pairwise correlation estimation and monitoring, including clustering (i.e., the dual to the outlier detection problem), novelty and intrusion detection among others.

The contributions of this work are:

- We introduce a suite of outlier detection techniques which are capable of operating over uni- or multidimensional outlier definitions as well as tumble or sliding streaming window paradigms along with combinations of them. We, thus end up with four modes of operation readily available to a wide class of application needs.
- For each operational mode, we introduce algorithms for data handling and transformation that preserve the ability to trade bandwidth consumption for accuracy during the outlier detection procedure. Moreover, our methods possess the ability to employ various useful similarity measures in our generic outlier definition.
- We present an extensive theoretic analysis on the quality guarantees that can be provided during the outlier detection procedure for each operational mode which is in direct relation to the desired degree of bandwidth consumption decrement.
- We conduct a detailed experimental analysis using a variety of real datasets and parameter settings. Our evaluation shows that our techniques remain accurate with average accuracy revolving around 80% in most of the cited cases and often exceeding 90%, while ensuring communication savings above 2.5 and up to a factor of 20.

Roadmap. The rest of the paper is organized as follows. In Section 2 we present preliminary concepts related to the sensor network architecture and mote data organization as well as the employed uni- and multidimensional outlier definitions. We also reason about the variety of similarity measures that are supported by our techniques. Section 3 presents the unidimensional outlier definition and tumbling window operational mode which is the only one supported by prior work [4, 5]. Section 4 details multidimensional outlier detection and tumble operation, unidimensional outlier detection and sliding window operation and multidimensional outlier detection and sliding window operation introduced in the current work with appropriate theoretic analysis. Related work is discussed in Section 5, before illustrating the results of our experimental evaluation in Section 6. Finally, Section 7 elaborates on the applicability of our techniques in a spectrum of IoT scenarios with privacy/security requirements and Section 8 includes concluding remarks.

2. Preliminaries

2.1. Network and Data Organization

We adopt an underlying network structure where motes are organized into clusters (shown as dotted circles in Figure 1) using any existing network clustering algorithm [14, 13]. An outlier detection query is propagated by a base station to the clusterheads, which, in turn, disseminate these queries to sensors within their cluster.

We assume that each mote in the network samples values for ξ monitored attributes and the outlier detection query has specified that decisions should be based to the ω most recent measurements obtained by motes. Consequently, the data organization at every sensor node is in the form of an $U^i \in R^{\xi \times \omega}$ matrix as depicted in Table 1. Columns $c_1, \cdots, c_\omega$ are ordered based on the time that attribute values were sampled by the corresponding sensories of mote S_i i.e. c_1 contains the most aged observations of attributes while c_ω includes the latest samples. Obviously, motes also employ a common order for the rows of U^i such that the $r - th$ row of the matrix corresponds to the same monitored attribute across the sensor network participants.

$$U^i = \begin{array}{c} \\ u_1 \\ u_2 \\ \vdots \\ u_\xi \end{array} \begin{array}{c} \begin{array}{cccc} c_1 & c_2 & \cdots & c_\omega \end{array} \\ \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1\omega} \\ u_{21} & \ddots & \vdots & u_{2\omega} \\ \vdots & \vdots & \ddots & \vdots \\ u_{\xi 1} & u_{\xi 2} & \cdots & u_{\xi\omega} \end{bmatrix} \end{array}$$

Table 1: Data organization at the sensor level. Each mote S_i maintains a matrix U^i comprised of ξ rows for the $u_1, \cdots, u_\xi$ monitored attributes and ω columns for the most recent measurements contained in the specified window size.

2.2. Outlier Definition

As in [2, 4, 5, 3, 6], we do not aim to compute outliers based on motes' latest reading but, instead, take into consideration their most recent measurements contained in a window of ω size. We first formally define the notion of an outlier for the unidimensional case [4, 5].

2.2.1. Unidimensional Outlier Definition

Since unidimensional outlier definition focuses on each monitored attribute separately, the ω most recent measurements of the $r-$th attribute (row in Table 1) kept by a mote S_i form a vector $u_r^i \in R^\omega$. Given a similarity metric $sim{:}R^\omega \times R^\omega \rightarrow [0, 1]$ and a *similarity threshold* ϕ_r we consider the readings by motes S_i and S_j with respect to their $r - th$ attribute similar if

$$sim(u_r^i, u_r^j) > \phi_r. \tag{1}$$

Overall, we classify a mote as an outlier if its $u_r^i \in R^\omega$ vector is not found to be similar with the corresponding vectors of at least $minSup_r$ other motes in the network.

The above, is a popular unidimensional outlier definition for sensor networks also employed in [2, 4, 5, 3, 6]. The utilization of the $minSup_r$ parameter renders the outlier detection techniques built upon the above definition resilient to cases when spurious measurements or totally abnormal observations simultaneously appear in the windows of multiple sensors.

2.2.2. Multidimensional Outlier Definition

We then formulate the generalization of the previous definition for the multidimensional case. Assuming that each mote S_i obtains samples for ξ monitored attributes, U^i denotes the corresponding matrix with ξ rows and ω columns (the window size) as in Table 1. For a similarity metric $sim{:}R^{\xi\times\omega} \times R^{\xi\times\omega} \rightarrow [0, 1]$ and a *s*imilarity threshold Φ, we consider the readings by motes S_i and S_j similar if

$$sim(U^i, U^j) > \Phi. \tag{2}$$

As with the unidimensional case, we classify a mote as an outlier if its U^i matrix is not found to be similar with the corresponding matrices of at least *minSup* other motes in the network. Notice that for $\xi = 1$ the latter definition reduces to the single attribute case, but we choose to make a distinction between them for clarity and ease of exposition. Moreover, we note that in both uni- or multi-dimensional outlier definitions, the minimum support parameter can be expressed either as an absolute number of motes or a percentage of nodes in the clusters.

2.3. Similarity Measures

Having formally presented the definition of an outlier in our setting, we concentrate on the particular *sim* measures that can be utilized in order to determine the similarity between pairs of motes in Inequalities 1 and 2. An important observation is that, in a sensor network environment, correlations among sensor measurements are more important than the absolute numbers of sampled values [2, 4]. For instance, considering motes that observe physical quantities such as noise levels, temperature etc; the absolute values of the acquired readings depend, for example, on the distance of a mote from the cause of the monitored event i.e., a passing car or a fire respectively. Thus, correlations among readings in space and time are more important than the absolute values [2, 4, 6].

Similarity Metric	Calculation of Similarity
Cosine Similarity	$cos(\theta(u_r^i, u_r^j)) = \frac{u_r^i \cdot u_r^j}{\|u_r^i\| \cdot \|u_r^j\|}$ $\Leftrightarrow \theta(u_r^i, u_r^j) = arccos \frac{u_r^i \cdot u_r^j}{\|u_r^i\| \cdot \|u_r^j\|}$
Correlation Coefficient	$corr(u_r^i, u_r^j) = \frac{cov(u_r^i, u_r^j)}{\sigma_{u_r^i} \sigma_{u_r^j}} =$ $= \frac{E(u_r^i u_r^j) - E(u_r^i)E(u_r^j)}{\sqrt{E(u_r^{i2}) - E^2(u_r^i)} \sqrt{E(u_r^{j2}) - E^2(u_r^j)}}$
Euclidean Distance	$dist(u_r'^i, u_r'^j) = \|u_r'^i - u_r'^j\|$

Table 2: Computation of some supported similarity metrics for unidimensional outlier detection. Vectors u_r^i, u_r^j contain the latest ω measurements of nodes S_i and S_j. In the table $E(.)$, σ, cov refer to the mean, the standard deviation and the covariance, respectively.

Similarity Metric	Calculation of Similarity
Inner Product	$R_{IN}(U^i, U^j) = \frac{Tr(U^{iT} \cdot U^j)}{\|U^i\|_F \cdot \|U^j\|_F}$
RV Coefficient	$RV(U^i, U^j) = \frac{Tr(U^i U^{iT} \cdot U^j U^{jT})}{\|U^i \cdot U^{iT}\|_F \cdot \|U^j \cdot U^{jT}\|_F}$
Matrix Distance	$R_F(U^i, U^j) = \|\frac{U^i U^{iT}}{\|U^i \cdot U^{iT}\|_F} - \frac{U^j U^{jT}}{\|U^j \cdot U^{jT}\|_F}\|_F$

Table 3: Generalization of similarity measures for matrix data. Matrices U^i, U^j contain the latest ω measurements of nodes S_i and S_j for ξ monitored attributes. In the latter table, $Tr(A)$ constitutes the trace of a matrix A, while A^T is the conjugate transpose of A. Additionally, $\|A\|_F$ stands for the Frobenius norm of A defined as $\|A\|_F = \sqrt{Tr(A \cdot A^T)}$.

Motivated by the above fact, our framework encompasses a wide variety of similarity measures that are able to capture such correlations [4, 5] in its unidimensional outlier detection operation (i.e. using Inequality 1 and the minimum support parameter). Table 2 presents some of the supported similarity metrics regarding the *Cosine Similarity*, the *Correlation Coefficient* and the *Vector (Euclidean) Distance*. Note that in the table $E(.)$, σ, cov refer to the mean, the standard deviation and the covariance, respectively. Furthermore, note that due to our previous discussion in the case of the vector distance, $u_r'^i$, $u_r'^j$ refer to the normalized versions of u_r^i, u_r^j, since as proven in [5] the Euclidean distance of normalized vectors enables us to capture the correlation among sensor readings and simultaneously allows the incorporation of distance based outlier definitions. These similarity measures (and also others discussed below) are supported by our solutions for the unidimensional outlier definition case and for both tumble as well as sliding window operational modes.

As regards our multidimensional outlier definition (Inequality 2), Table 3 lists versions of the previously discussed measures able to capture matrix correlations [17, 18]. In particular, the *Inner Product Correlation* constitutes a generalization of the Cosine similarity [17], the *RV-Coefficient* is a multidimesnional version of the Correlation Coefficient [18], while the *Matrix Distance* refers to measuring the relative (dis)similarity between matrices [18]. In the latter table, $Tr(A)$ constitutes the trace of a matrix A, while A^T is the conjugate transpose of A. Additionally, $\|A\|_F$ stands for the Frobenius norm of A defined as $\|A\|_F = \sqrt{Tr(A \cdot A^T)}$.

In what follows we will concentrate on the Cosine Similarity and Inner Product Correlation cases as running examples for the uni- and multi-dimensional outlier detection, respectively. That is because: a) when considering unidimensional outlier definitions $cos(\theta(u_r^i, u_r^j))$ serves as the building block for encompassing the other measures noted in Table 2 in the outlier detection process (see [5]) and b) in Table 3 it is evident that the $RV(U^i, U^j) = R_{IN}(U^i \cdot U^{iT}, U^j \cdot U^{jT})$ and also $R_F(U^i, U^j) = \sqrt{2} \cdot \sqrt{1 - RV(U^i, U^j)}$ holds [18].

It is important to emphasize that the framework is not limited to using just the triplet of similarity measures presented in the tables. In [5] it is shown that similarity measures like the *Tanimoto Coefficient* and the *Jaccard Index* can also be used and remain applicable to the operational modes that are introduced in the current work for unidimensional outlier detection. In addition, [17] exhibits how a number of other (e.g. spectra or orientation independent) matrix correlation measures can be expressed in an R_{IN} form and are thus commutatively supported in multidimensional outlier definitions and the corresponding operational modes presented here.

Symbol	Description
S_i	the $i-th$ sensor node.
ω	window (tumble or sliding) size.
ξ	number of monitored mote attributes.
d	bitmap length.
U^i	the matrix of $\xi \times \omega$ dimensionality containing the values of all monitored attributes of mote S_i.
u^i_r	the value vector of the $r-$th attribute ($r-$th row in U^i) of mote S_i.
c^i_k	the $k-$th column in U^i of mote S_i.
u^i_{rk}	the value of the cell at the $r-$th row and $k-$th column in U^i of mote S_i.
$\theta(x, y)$	the angle between vectors x, y.
X^i, X^i_u	the bitmap encoding produced after applying LSH to matrix U^i and a vector u (respectively) of mote S_i.
$D_h(X, Y)$	the hamming distance between a pair of bitmaps X, Y.
Φ, Φ_θ	cosine and angle similarity threshold (respectively) in the multidimensional case.
ϕ_r, ϕ_{r_θ}	cosine and angle similarity threshold (respectively) in the unidimensional case ($r-th$ attribute).
Φ_{D_h}, $\phi_{r_{D_h}}$	similarity threshold based on hamming distance for the multi- and unidimensional ($r-th$ attribute) case, respectively.
Sup^i, Sup^i_r	the support of mote S_i for the multi- and unidimensional ($r-th$ attribute) case, respectively.
$minSup$, $minSup_r$	the minimum support parameter for the multi- and unidimensional case, respectively.
λ	batch size ($1 \leq \lambda \leq \omega$) used in SWO-UnO operational mode.
$u^i_{rb_k}$	the batch at the $r-$th row (attribute) in U^i of size λ, that ends at the $k-$th column of U^i. Used in SWO-UnO operational mode.

Table 4: Frequently Used Symbols

3. Basic TACO Framework

In this section we exhibit the operation of the TACO framework for the unidimensional outlier detection case utilizing a tumble of measurements. Recall that, as we noted in our introductory section, tumbles are defined as disjoint windows [15] of attribute samples and as a result every row of the U^i matrix (shown in Table 1) expires altogether so as to be filled with new measurements for the next tumbles of attributes. We mention beforehand, that this is the only operational mode that is supported by [4, 5].

3.1. Tumble Window Operation - Unidimensional Outlier Detection (TWO-UnO) Operational Mode

3.1.1. TWO-UnO Processing at the Sensor Level

At a first step, TWO-UnO applies LSH to the value vectors of quantities sampled by motes. LSH [10] schemes have the property of dimensionality reduction while preserving similarity between vectors under comparison. In our setting, for each monitored attribute $u^i_r \in R^\omega$, mote S_i encodes its latest ω measurements using a bitmap of $d \ll \omega \cdot B(u^i_{rk})$ bits, where $B(u^i_{rk})$ denotes the size of the binary representation of a single measurement. In particular, the performed data compression is based on the Random Hyperplane Projection (RHP) LSH scheme, formally presented in the following theorem.

Theorem 1. *[Random Hyperplane Projection (RHP) [10, 19]]*
Assume we are given a collection of vectors defined on the ω dimensional space. We choose a family of hash functions as follows: We produce a spherically symmetric random vector ρ of unit length from this ω dimensional space. We define a hash function h_ρ as:

$$h_\rho(u^i_r) = \begin{cases} 1 & ,if\ \rho \cdot u^i_r \geq 0 \\ 0 & ,if\ \rho \cdot u^i_r < 0 \end{cases}$$

For any two vectors $u_r^i, u_r^j \epsilon R^\omega$:

$$P = P[h_\rho(u_r^i) = h_\rho(u_r^j)] = 1 - \frac{\theta(u_r^i, u_r^j)}{\pi} \square \tag{3}$$

Equation 3 can be rewritten as:

$$\theta(u_r^i, u_r^j) = \pi \cdot (1 - P) \tag{4}$$

Note that Equation 4 expresses angle similarity as the product of the potential range of the angle between the two vectors (π), with the probability of equality in the result of the hash function application (P). Thus, after repeating a stochastic procedure using d random vectors ρ, the final embodiment in the hamming cube results in [20]:

$$D_h(X_{u_r}^i, X_{u_r}^j) = d \cdot (1 - P) \tag{5}$$

where $X_{u_r}^i, X_{u_r}^j \in [0,1]^d$ are the bitmaps (of length d) produced and $D_h(X_{u_r}^i, X_{u_r}^j) = \sum_{\ell=1}^{d} |X_{u_{r\ell}}^i - X_{u_{r\ell}}^j|$ is their hamming distance. Hence, we finally derive:

$$\frac{\theta(u_r^i, u_r^j)}{\pi} = \frac{D_h(X_{u_r}^i, X_{u_r}^j)}{d} \tag{6}$$

Having encapsulated a common *seed* parameter in the outlier detection query, each mote S_i is able to produce the same d random vectors ρ, determine the d signs of $\rho \cdot u_r^i$ inner products as devised by Theorem 1 and come up with a bitmap $X_{u_r}^i \in [0,1]^d$ for its u_r^i attribute. This compressed bitmap representation is then transmitted to the clusterhead that S_i belongs.

3.1.2. TWO-UnO Intracluster Processing

Upon the receipt of the bitmaps of every mote within its cluster, a clusterhead C_i is to obtain similarity estimations for the $r-th$ monitored attribute. Equation 6 provides the means to compute the angle (and thus the cosine similarity) between the initial value vectors based on the hamming distance of their corresponding bitmaps. Moreover, according to Equation 6 the posed similarity threshold ϕ_{r_θ} can be expressed in terms of a hamming distance threshold as $\phi_{r_{D_h}} = d\frac{\phi_{r_\theta}}{\pi}$. As a consequence, C_i performs pairwise similarity checks and classifies two motes as similar if $D_h(X_{u_r}^i, X_{u_r}^j) \leq \phi_{r_{D_h}}$ in which case it also increases the support of the corresponding motes by one i.e. $Sup_r^i \leftarrow Sup_r^i + 1$ and $Sup_r^j \leftarrow Sup_r^j + 1$ is performed. Afterwards, C_i places motes that do not reach the required $minSup_r$ in a list of potential outliers $PotOut_{C_i}$ in the form of $\langle S_i, X_{u_r}^i, Sup_r^i \rangle$ triplets. Note that these potential outliers are expected to be only a small fraction of motes participating in a cluster.

3.1.3. TWO-UnO Intercluster Processing

The extracted $PotOut_{C_i}$ lists of motes with $Sup_r^i < minSup_r$ are not final outliers since the outlier detection query may have allowed motes in different clusters to be tested for similarity as well. For instance, users may allow motes within a certain radius to be able to witness each other, irrespectively of whether they have been assigned to the same cluster, as they are expected to be able to sense similar conditions. As another example, users may have specified that only readings from vibration sensors attached on identical (but distant to be assigned to the same clusterhead) engines in a machine room are comparable. Motes in the lists extracted by cluster C_i that are not subjected to such kind of specifications can be directly reported to the query source. Otherwise, $\langle S_i, X_{u_r}^i, Sup_r^i \rangle$ triplets need to be communicated among clusters. Given the current cluster as the starting node, query-specified clusterheads as intermediate sites and the base station for the destination, the intercluster communication problem is modeled as a TSP according to which $PotOut_{C_i}$s are exchanged between clusterheads participating in the path. The TSP problem can be solved either by the base station after clusterhead election or adopting a greedy distributed algorithm that forwards potential outliers to the nearest cluster which has not been visited yet. The latter routing paradigm can be directly supported by utilizing a geographic aware routing protocol such as GPSR [21] at the clusterheads level. Every $\langle S_i, X_{u_r}^i, Sup_r^i \rangle \in PotOut_{C_i}$ that manages to reach $minSup_r$ is excluded from the list that will be forwarded to the next clusterhead, while if it does not it will be finally reported to the basestation. The described procedure is repeated upon the formation of a new tumble.

3.1.4. TWO-UnO Analysis

As regards the quality guarantees that are provided by the framework, we initially focus on the ability of the estimated $\frac{D_h(X^i_{u_r}, X^j_{u_r})}{d}$ to approximate the true value of $\frac{\theta(u^i_r, u^j_r)}{\pi}$ in Equation 6. The following theorem relates the accuracy of the approximation to the bandwidth consumption that the application is willing to sacrifice.

Theorem 2. *[5] To estimate $\frac{\theta(u^i_r, u^j_r)}{\pi}$ with precision ϵ and probability at least $1-\delta$ ($0 < \epsilon, \delta < 1$) using Equation 6, sensor nodes need to produce bitmaps of $O(\ell og(2/\delta)/(2\epsilon^2))$ length.*

By keeping the estimated hamming distance close to its expected value (Equation 6), it is more likely to perform a correct similarity test. More precisely, we want to quantify the probability that if u^i_r, u^j_r are similar $\theta(u^i_r, u^j_r) \leq \phi_{r_\theta}$ then $D_h(X^i_{u_r}, X^j_{u_r}) \leq \phi_{r_{D_h}}$ and accordingly for dissimilar vectors.

Theorem 3. *[5] For $\theta(u^i_r, u^j_r) > 0$ and for bitmaps of $d = O(\ell og(2/\delta)/(2\epsilon^2))$ length (Theorem 2), clusterheads perform a correct similarity test by means of $D_h(X^i_{u_r}, X^j_{u_r})$ (Equation 6) with probability at least $1 - e^{-\frac{(\theta(u^i_r, u^j_r) - \phi_{r_\theta})^2}{\theta(u^i_r, u^j_r)} \frac{d}{2\pi^2}}$.*

The above theorem essentially states that the probability of an incorrect similarity test at a clusterhead using TWO-UnO decreases exponentially with $|\theta(u^i_r, u^j_r) - \phi_{r_\theta}|$. Note that when $\theta(u^i_r, u^j_r) = 0$ the produced bitmaps will be identical and thus the framework can always correctly classify u^i_r, u^j_r as similar [5].

Bandwidth vs Accuracy Trade off. According to Theorem 2, the tighter the quality guarantees of the framework become, i.e., both (ϵ, δ) decrease; the size of the bitmaps, proportional to $\ell og(2/\delta)/(2\epsilon^2)$, increases. This exhibits how TWO-UnO straightforwardly trades bandwidth usage for estimation accuracy. Furthermore, according to Theorem 3, the error probability δ of a similarity test steeply decreases for lower ϕ_{r_θ} values. This exhibits that the technique is expectedly more accurate with stricter outlier definitions expressed by lower ϕ_{r_θ}.

4. Our Omnibus Outlier Detection Solution and Quality Guarantees

We now proceed to the introduction of our algorithmic suite as described in Section 1. In this section, we (a) enable support for multidimensional outlier detection upon operating on a per tumble fashion, (b) introduce multidimensional outlier detection techniques for a sliding window paradigm and (c) provide the ability for unidimensional outlier detection under sliding windows, as well. We choose to present our algorithms using the above order for ease of exposition and to better exhibit the flow behind the fostered rationale.

4.1. Tumble Window Operation - Multidimensional Outlier Detection (TWO-MuO) Operational Mode

In Section 2 we presented a formal definition based on which multidimensional outlier detection is to take place within the network setting, while we also reasoned about focusing on the R_{IN} (see Table 3) matrix correlation metric. The first challenge that we need to confront in our effort to incorporate multidimensional outlier definitions in our framework comes from the fact that the nature of the data that need to be encoded and subsequently compared for similarity is now different. In particular, instead of vectors as in Section 3, we now face the situation of comparing matrices. More importantly, Theorem 1 can be applied to vectors instead of matrices and is designed to preserve the angle (and thus the cosine) similarity rather than R_{IN}.

Let Vec be an isomorphism such that $Vec : R^{\xi \times \omega} \rightarrow R^{\xi\omega}$. This essentially is a linear transformation that renders matrices to vectors by stacking the columns of any processed matrix. According to the data organization in Table 1, at a mote S_i, column c_1 is placed on the top of the stack while $c_2, \cdots, c_\omega$ are appended to it forming a single vector of $\xi \cdot \omega$ dimension, that is $Vec(U^i)$. Let us focus on the elements involved in R_{IN} calculation (see Table 3), starting with the numerator $Tr(U^{iT} \cdot U^j)$. The proposition below shows how the trace of matrix multiplication can be equivalently transformed to the inner product of their Vec form. Notice that in our setting motes possess $R^{\xi \times \omega}$ matrices and as such they are always of proper dimension.

Proposition 1. *The trace of the product of two properly dimensioned matrices, equals the inner product of their Vec forms. In our setting: $Tr(U^{iT} \cdot U^j) = Vec(U^i) \cdot Vec(U^j)$.*

PROOF. The trace of the matrix yielded by $U^{iT} \cdot U^j$ is the sum of its diagonal elements. If we symbolize u_k^{iT} the k-th row of U^{iT} and c_k^j the k-th column of U^j then $Tr(U^{iT} \cdot U^j) = \sum_{k=1}^{\omega} u_k^{iT} \cdot c_k^j = Vec(U^i) \cdot Vec(U^j)$. Formally, $Vec(U^i) \cdot Vec(U^j)$ is actually $Vec(U^i)^T \cdot Vec(U^j)$, but we omit the T sign from the vector as is common in inner product notation. □

We then concentrate on the denominator of R_{IN} computational formula (Table 3). The following corollary is a direct outcome of Proposition 1.

Colorrary 1. *The Frobenius norm of a matrix equals the L_2 norm of its Vec form. In our setting: $\|U^i\|_F = \|Vec(U^i)\|$.*

Based on Proposition 1 and Corollary 1:

$$R_{IN}(U^i, U^j) = cos(\theta(Vec(U^i), Vec(U^j))) \Leftrightarrow$$
$$R_{IN}^{\theta}(U^i, U^j) = arccos(R_{IN}(U^i, U^j)) = \theta(Vec(U^i), Vec(U^j)) \tag{7}$$

4.1.1. TWO-MuO Processing at the Sensor Level

In the previous paragraphs we achieved to express the $U^i \in R^{\xi \times \omega}$ matrix locally maintained at mote S_i as a $R^{\xi\omega}$ vector. As a result, the first step towards multidimensional outlier detection in a per tumble fashion, comes at the sensor level where every mote in the network applies the Vec operator and appropriately transforms acquired samples of data. Because of the transformation $Vec(U^i) \in R^{\xi\omega}$, Theorem 1 (which operates on vectors rather than matrices) is now applicable. Hence motes can utilize the RHP scheme to produce X^i bitmaps for U^i, knowing that at the next step clusterheads will indeed be able to deduce existing correlations using the bitmaps due to Equation 6 and Equation 7. This time, to allow for communication savings, the length of the bitmap should be chosen to be $d \ll \xi \cdot \omega \cdot B(u_{rk}^i)$ (instead of $d \ll \omega \cdot B(u_{rk}^i)$ in the unidimensional case). Subsequently, X^i bitmaps are transmitted to the corresponding clusterheads.

4.1.2. TWO-MuO Intra- & Intercluster Processing

Initially, by combining Equations 6 and 7 we have:

$$\frac{R_{IN}^{\theta}(U^i, U^j)}{\pi} = \frac{D_h(X^i, X^j)}{d} \tag{8}$$

Given an $R_{IN}-$similarity threshold Φ specified by the posed outlier detection query, clusterheads are able to transform this threshold to an equivalent $\Phi_\theta = arccos(\Phi)$ using Equation 7 and then to a corresponding hamming distance threshold $\Phi_{D_h} = d\frac{\Phi_\theta}{\pi}$ by Equation 8. That point forward the outlier detection procedure continuous as with the TWO-UnO case both during the intra- and intercluster processing phases. If $D_h(X^i, X^j) \leq \Phi_{D_h}$ then $Sup^i \leftarrow Sup^i + 1$, $Sup^i \leftarrow Sup^i + 1$ is performed, while motes that do not reach the necessary $minSup$ are placed in $PotOut_{C_i}$ lists. Recall, that since in this operational mode the outlier identification is still performed in a per tumble fashion, the whole U^i matrix (tumble) expires so as to be filled with new measurements and the procedure presented here is again applied on the upcoming tumbles.

4.1.3. TWO-MuO Analysis

Lemma 1 elaborates on the accuracy of the approximation of the above Equation 8 generalizing Theorem 2.

Lemma 1. *$\frac{R_{IN}^{\theta}(U^i, U^j)}{\pi}$ can be estimated with precision ϵ and probability at least $1 - \delta$ using Equation 8, when sensor nodes produce bitmaps of $O(\ell og(2/\delta)/(2\epsilon^2))$ length.*

Besides, as regards the accuracy of the performed similarity tests, Lemma 2 adjusts Theorem 3 for the multidimensional outlier detection - tumble operation case.

Lemma 2. *For $R_{IN}^{\theta}(U^i, U^j) > 0$ and for bitmaps of $d = O(\ell og(2/\delta)/(2\epsilon^2))$ length (Lemma 1), clusterheads perform a correct similarity test by means of $D_h(X^i, X^j)$ (Equation 8) with probability at least $1 - e^{-\frac{(R_{IN}^{\theta}(U^i,U^j) - \Phi_\theta)^2}{R_{IN}^{\theta}(U^i,U^j)} \frac{d}{2\pi^2}}$.*

The proofs are straightforward according to our discussion in this subsection and are omitted. Again notice that the above lemma states that the probability of incorrect estimation in the TWO-MuO operational mode decreases exponentially with $|R^{\theta}_{IN}(U^i, U^j) - \Phi_\theta|$.

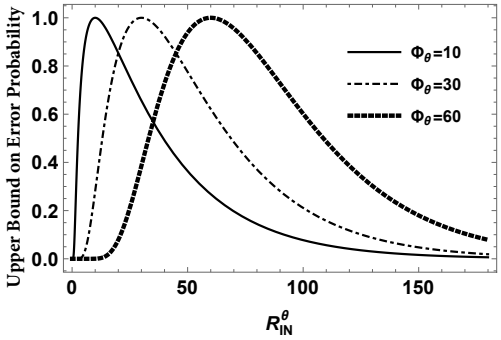


Figure 2: TWO-MuO upper bound on the probability of erroneous similarity test varying $R^{\theta}_{IN}, \Phi_\theta$

Bandwidth vs Accuracy Trade off. Again, based on Lemma 1, the tighter the quality guarantees of the framework become, i.e., both (ϵ, δ) decrease; the size of the bitmaps, proportional to $\ell og(2/\delta)/(2\epsilon^2)$, increases. This exhibits that TWO-MuO trades bandwidth for accuracy similarly to TWO-UnO. Furthermore, according to Lemma 2, the value that bounds the probability of an incorrect similarity test, decreases more steeply for lower Φ_θ values. Figure 2 illustrates the values of the upper bound on the probability of an incorrect similarity test at the clusterhead level, based on Lemma 2 (vertical axis) varying R^{θ}_{IN} for three different cases of Φ_θ. It can easily be observed that the surface covered by each curve corresponding to 10, 30, 60 degrees of angle similarity thresholds, increases with Φ_θ. Hence, TWO-MuO is expectedly more accurate with stricter outlier definitions expressed by low Φ_θs.

TWO-MuO Trade off Practical Example. In case of picking a negligible δ (for instance $\delta = 0.001$) and $\epsilon = 0.05$, the proper bitmap size is a fraction (e.g. 1/2, 1/4, 1/8, 1/16 etc see [4, 5]) of *maximum* 83 bytes, which involves the consumed bandwidth for S_i per tumble. For instance, for $|R^{\theta}_{IN}(U^i, U^j) - \Phi_\theta| = 0.35$ rad and for $R^{\theta}_{IN}(U^i, U^j) = \pi/12$ ($\sim$ 15 degrees), the probability of an error in a similarity test is bounded by $1.46 \cdot 10^{-7}$.

4.2. Sliding Window Operation - Multidimensional Outlier Detection (SWO-MuO) Operational Mode

Upon switching to a sliding window operation, the major difference in our outlier detection computational model comes from the fact that U^i matrices do not expire altogether as with the paradigms presented so far. In the new setting, as soon as a new sample for each monitored attribute is obtained, it is appended to the U^i matrix as a new column, while the first column (containing the most aged samples, Section 2.1) expires. An instance of this process is depicted in Table 5.

Slide $\Rightarrow$				Expired	New Window (column $\downarrow$)			
10	12.2	$\cdots$	18.3	10	12.2	$\cdots$	18.3	8.75
21	$\ddots$	$\vdots$	28.6	21	$\vdots$	$\ddots$	28.6	4
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
31.9	32.1	$\cdots$	38.3	31.9	32.1	$\cdots$	38.3	31

Table 5: Exemplary exposition of the changes S_i's data undertake when operating in a sliding window paradigm.

Assume that we do not impose any compression on the sampled values in U^i. At the beginning of the outlier detection query operation, every mote fills in U^i with acquired samples since it is initially empty, and transmits this

information to its clusterhead. For the subsequent slides of the window, however, each mote S_i needs to only transmit the lastly introduced column which after the window slide is c_ω. The clusterheads are then capable of recognizing that the first column of the old window has expired so as to drop it, slide the columns as in Table 5 and append the new c_ω to it.

Considering our need to reduce bandwidth consumption, the major issue that arises is that now motes cannot compress the matrices at once as they did in Section 4.1 and estimate R^{θ}_{IN} using $D_h(X^i, X^j)$. This is due to the fact that at any given time, the new part that needs to be processed involves an individual column. If motes are to compress something to ensure communication savings, this can only be the new c_ω. On the other hand, if they do so, a natural question that arises regards whether we will still be able to preserve the R_{IN} similarities of the whole U_i using compression of single columns. It is easy to verify that the X^i bitmap that motes locally produced by applying RHP on the whole U^i cannot be derived at the clusterhead level if clusterheads receive bitmaps yielded from RHP application on the new c_ω. This is true even when trying to concatenate those bitmaps or apply a binary operator (e.g. bitwise OR/AND) on them, because the final signs of the inner products of Theorem 1 cannot be safely determined to be identical to those computed having the whole U^i available.

To confront this situation, achieve bandwidth consumption reduction (through data compression) and similarity preservation taking advantage of the RHP scheme, we need to reconsider our strategies. We first formulate the operation of the outlier detection algorithm for the multidimensional outlier detection - sliding window operational mode and subsequently we elaborate on the quality guarantees that are attributed to our framework utilizing the aforementioned algorithm. In Equation 7 we showed that $R_{IN}(U^i, U^j) = cos(Vec(U^i), Vec(U^j))$, so our algorithms focus on preserving the cosine similarity of the Vec forms of the corresponding matrices.

4.2.1. SWO-MuO Processing at the Sensor Level

After obtaining new samples for the ξ monitored attributes, mote S_i shifts the window (Table 5) and appends the new c^i_ω column to it (the superscript i again stands for the S_i mote). Since the rest of the columns have already been processed and communicated to the clusterheads (using the same procedure which we are going to present hereafter), we concentrate on the local data processing the new c^i_ω column (after each and every shift of the sliding window) should undertake. At the first step of the algorithm S_i applies the RHP LSH scheme on the $c^i_\omega \in R^\xi$ vector and derives a bitmap $X^i_{c_\omega}$ of d length. Moreover, it computes $\|c^i_\omega\|$ i.e. the L_2 norm of the values in the c^i_ω vector and transmits this information together with $X^i_{c_\omega}$ to the clusterhead. Let $B(\|c^i_\omega\|)$ the size of the binary representation of the L_2 norm (usually 32 bits). Then, to achieve compression, we should ensure that $(d + B(\|c^i_\omega\|))$ is (much) smaller than the size of the initial ξ monitored attributes. This can be tuned, as discussed in our subsequent analysis, for motes monitoring $\xi > 2$ attributes and consequently our techniques are capable of ensuring compression in any practical scenario.

4.2.2. SWO-MuO Intracluster Processing

As soon as clusterheads receive $< X^i_{c_\omega}, \|c^i_\omega\| >$ from motes in their cluster, they need to perform the required similarity checks. For any pair of matrices U^i, U^j under comparison, a clusterhead C_i needs to estimate the value of $R_{IN}(U^i, U^j) = cos(Vec(U^i), Vec(U^j)) = \frac{Vec(U^i)\cdot Vec(U^j)}{\|Vec(U^i)\|\cdot\|Vec(U^j)\|}$ (Equation 7). Initially, notice that by having motes send $\|c^i_\omega\|$, $\|c^j_\omega\|$, clusterheads derive the exact value for $\|Vec(U^i)\| = \sqrt{\|c^i_1\|^2 + \cdots + \|c^i_\omega\|^2}$ since as previously mentioned $\|c^i_1\|, \cdots, \|c^i_{\omega-1}\|$ have been received at the previous shifts of the window. The same holds for $\|Vec(U^j)\|$.

Secondly, due to the fact that the inner product between vectors is a distributive operator and recalling that the application of the Vec operator creates an ordered stack of columns (Section 4.1):

$$Vec(U^i) \cdot Vec(U^j) = c^i_1 \cdot c^j_1 + \cdots + c^i_\omega \cdot c^j_\omega \tag{9}$$

and clusterhead C_i is able to derive an estimation of

$$c^i_\omega \cdot c^j_\omega = cos(\theta(c^i_\omega, c^j_\omega)) \cdot \|c^i_\omega\| \cdot \|c^j_\omega\| \tag{10}$$

utilizing the received bitmaps. More precisely, if $\widehat{c^i_\omega \cdot c^j_\omega}$ denotes the estimation of $c^i_\omega \cdot c^j_\omega$, it can be computed using the received $X^i_{c_\omega}, X^j_{c_\omega}$ bitmaps as:

$$\widehat{c^i_\omega \cdot c^j_\omega} \overset{Eq.\ 6\ \&\ 10}{=} cos(\frac{D_h(X^i_{c_\omega}, X^j_{c_\omega})}{d}\pi) \cdot \|c^i_\omega\| \cdot \|c^j_\omega\| \tag{11}$$

Then, considering that also $\widehat{c_k^i \cdot c_k^j}$ for $1 \leq k < \omega$ have been computed at the previous window slides, the overall estimation $\widehat{Vec(U^i) \cdot Vec(U^j)}$ can be derived using the extracted partial estimations and Equation 9 as $\widehat{Vec(U^i) \cdot Vec(U^j)} = \widehat{c_1^i \cdot c_1^j} + \cdots + \widehat{c_\omega^i \cdot c_\omega^j}$. The final estimation of $R_{IN}(U^i, U^j) = cos(Vec(U^i), Vec(U^j))$ is:

$$cos(\theta(\widehat{Vec(U^i), Vec(U^j)})) = \frac{\widehat{Vec(U^i) \cdot Vec(U^j)}}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} \tag{12}$$

The common part regarding $cos(\widehat{Vec(U^i), Vec(U^j)}) \geq \Phi$ similarity tests is then performed at the intracluster processing phase and motes that do not reach the required $minSup$ are added to the $PotOutC_i$ list. Nevertheless, as we will present in the next paragraph, this time the contents of $PotOutC_i$ depend on how often a mote takes place in the intercluster communication phase.

4.2.3. SWO-MuO Intercluster Processing

During the intercluster communication phase, motes included in the $PotOut_{C_i}$ list are communicated between clusterheads following computed TSP paths as in Section 3. However, in Section 3 the utilized operational mode was functioning under tumbled window adoption, while the same holds for multidimensional outlier detection in Section 4.1. The current operational mode entails support for sliding windows and multidimensional outlier definitions and thus the information that needs to be communicated when a mote participates in the intercluster processing phase depends on the frequency with which it is included in $PotOut_{C_i}$. In particular, assume that at the current instance a mote S_i is included in $PotOut_{C_i}$ for the first time. The ω bitmaps $X^i_{c_1}, \cdots, X^i_{c_\omega}$ for each column that have been communicated to C_i from S_i need to be included in $PotOut_{C_i}$ together with the respective norms i.e. a triplet $< S_i, \{< X^i_1, \|c^i_1\| >, \cdots, < X^i_{c_\omega}, \|c^i_\omega\| >\}, Sup^i >$. This triplet will be forwarded to the next clusterhead in the TSP path. Notice that should we had uncompressed data the triplet would be $< S_i, \{c^i_1, \cdots, c^i_\omega\}, Sup^i >$ and hence, in the multidimensional case, bandwidth preservation through $X^i_{c_1}, \cdots, X^i_{c_\omega}$ utilization is achieved in this processing phase as well. Nonetheless, if S_i is again placed in $PotOut_{C_i}$ after $1 \leq \tau < \omega$ shifts of the window, C_i needs to shorten the latter triplet to $< S_i, \{< X^i_{c_{\omega-\tau}}, \|c^i_{\omega-\tau}\| >, \cdots, < X^i_{c_\omega}, \|c^i_\omega\| >\}, Sup^i >$ since the rest of the bitmaps were communicated among clusterheads at the initial S_i inclusion in $PotOut_{C_i}$.

Notice that the same approach is applied not only to the clusterhead that is the origin of the TSP path, but also for every intermediate cluster where the value of τ may differ for each clusterhead pair on a respective TSP edge. This is because if S_i reaches a clusterhead where it finds adequate support and thus Sup^i surpasses $minSup$, it will not be forwarded to the next hop in the path. Since each cluster knows the number of window slides since the last intercluster communication with its TSP neighbor, it can compute τ in any such case and properly form the $< S_i, \{< X^i_{c_{\omega-\tau}}, \|c^i_{\omega-\tau}\| >, \cdots, < X^i_{c_\omega}, \|c^i_\omega\| >\}, Sup^i >$ triplet. That aside, the intercluster communication phase proceeds as usual and motes that manage to accumulate $\geq minSup$ support are excluded from the list that will be routed to the next clusterhead. Eventually, the final outliers (those with $Sup^i < minSup$) are reported to the base station which is the final destination of the followed path.

4.2.4. SWO-MuO Analysis

We will first direct our analysis towards identifying the accuracy that $cos(\widehat{Vec(U^i), Vec(U^j)})$ in Equation 12 can provide with respect to the true value of $cos(Vec(U^i), Vec(U^j))$.

Theorem 4. *For bitmaps of $O(\ell log(2/\delta)/(2\epsilon^2))$ length (Theorem 2) the estimation of Equation 12 lies in:*

$$cos(\widehat{Vec(U^i), Vec(U^j)}) \in cos(Vec(U^i), Vec(U^j)) \pm 2sin(\frac{\pi}{2}\epsilon)$$

with probability at least $\Delta = (1-\delta)^\omega$.

PROOF. Having compressed bitmaps according to Theorem 2 we know that $\frac{D_h(X^i_{c_\omega}, X^j_{c_\omega})}{d}\pi = \widehat{\theta(c^i_\omega, c^j_\omega)} \in \theta(c^i_\omega, c^j_\omega) \pm \pi\epsilon$ with probability $1-\delta$. Moreover, the RHP application yields an angle similarity that lies in $0 \leq \theta(c^i_\omega, c^j_\omega) \leq \pi$. This means that $cos(\widehat{\theta(c^i_\omega, c^j_\omega)}) \in cos(\theta(c^i_\omega, c^j_\omega) \pm \pi\epsilon)$.

We then focus on the error that may occur when estimating the inner product of the $k-th$ pair of columns $c_k^i \cdot c_k^j$ using Equation 11. According to our previous analysis:

$$\begin{aligned}
|c_k^i \cdot c_k^j - \widehat{c_k^i \cdot c_k^j}| = \quad & |cos(\theta(c_k^i, c_k^j)) - cos(\theta(\widehat{c_k^i, c_k^j}))|\|c_k^i\| \cdot \|c_k^j\| \\
\in \quad & |cos(\theta(c_k^i, c_k^j)) - cos(\theta(c_k^i, c_k^j) \pm \pi\epsilon)|\|c_k^i\| \cdot \|c_k^j\| \\
(\text{trigonom. identity}) \in \quad & |-2sin(\theta(c_k^i, c_k^j) \pm \tfrac{\pi\epsilon}{2})| \cdot |sin(\pm\tfrac{\pi}{2}\epsilon)| \cdot \|c_k^i\| \cdot \|c_k^j\| \\
(|sin(\theta(c_k^i, c_k^j) \pm \tfrac{\pi\epsilon}{2})| \leq 1, sin(-\theta) = -sin(\theta)) \in \quad & 2|sin(\tfrac{\pi}{2}\epsilon)| \cdot \|c_k^i\| \cdot \|c_k^j\| \\
\widehat{c_k^i \cdot c_k^j} \in \quad & c_k^i \cdot c_k^j \pm 2sin(\tfrac{\pi}{2}\epsilon) \cdot \|c_k^i\| \cdot \|c_k^j\|
\end{aligned}$$

As stated in Theorem 2, the above happens with probability $1-\delta$. Now, we proceed aiming at bounding the accuracy provided by the $cos(\theta(\widehat{Vec(U^i)}, \widehat{Vec(U^j)}))$ in Equation 12. To do so, we assume that *the estimation error* (i.e., not the column values which can still bear dependences in every slide of the window) between $\widehat{c_k^i \cdot c_k^j}$ estimations of different pairs of columns is independent. In practice, this can be achieved by using the common seed that was transmitted on par with the outlier detection query (Section 3) to recursively produce common seeds in all motes upon compressing individual columns. Hence, the following bounds the accuracy of the estimation with probability $\Delta = (1-\delta)^\omega$. By Equation 9 and the analysis of the previous paragraph:

$$cos(\widehat{Vec(U^i), Vec}(U^j)) \in \frac{\sum_{k=1}^{\omega} c_k^i \cdot c_k^j \pm 2sin(\frac{\pi}{2}\epsilon) \cdot \sum_{k=1}^{\omega} \|c_k^i\| \cdot \|c_k^j\|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} =$$

$$\frac{\sum_{k=1}^{\omega} c_k^i \cdot c_k^j}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} \pm 2sin(\frac{\pi}{2}\epsilon) \frac{\sum_{k=1}^{\omega} \|c_k^i\| \cdot \|c_k^j\|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} =$$

$$cos(Vec(U^i), Vec(U^j)) \pm 2sin(\frac{\pi}{2}\epsilon) \frac{\sum_{k=1}^{\omega} \|c_k^i\| \cdot \|c_k^j\|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|}$$

By the Cauchy-Schwarz Inequality:

$$\frac{\sum_{k=1}^{\omega} \|c_k^i\| \cdot \|c_k^j\|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} = \frac{\sum_{k=1}^{\omega} \|c_k^i\| \cdot \|c_k^j\|}{\sqrt{\sum_{k=1}^{\omega} \|c_k^i\|^2 \cdot \sum_{k=1}^{\omega} \|c_k^j\|^2}} \leq 1$$

and finally

$$cos(\widehat{Vec(U^i), Vec}(U^j)) \in cos(Vec(U^i), Vec(U^j)) \pm 2sin(\frac{\pi}{2}\epsilon)$$

which concludes our proof. □

Notice that the smaller the error ϵ present in individual bitmaps, the closer to zero $\pm 2sin(\frac{\pi}{2}\epsilon)$ will be. Hence, the accuracy of the $cos(\widehat{Vec(U^i), Vec}(U^j))$ estimation increases. The above theorem refers to the accuracy of R_{IN}, via the cosine of the *Vec* forms, similarity estimation. Nevertheless, in our setting we are particularly interested in the accuracy of the performed similarity tests given a similarity threshold. The following theorem, exploits Theorem 4 focusing on the accuracy of such tests.

Theorem 5. *For bitmaps of $O(\ell log(2/\delta)/(2\epsilon^2))$ length (Theorem 2), clusterheads perform a correct similarity test upon utilizing $cos(\widehat{Vec(U^i), Vec}(U^j))$ (Equation 12) with probability at least* $1 - \min_{t>0} e^{-t\cdot|\Phi - cos(Vec(U^i), Vec(U^j))|}((1-\delta) \cdot e^{t\cdot 2|sin(\frac{\pi\epsilon}{2})|} + \delta \cdot e^{2\cdot t})^\omega$.

PROOF. Without loss of generality we assume that $cos(\ Vec(U^i), Vec(U^j)) \leq \Phi$. An error in the similarity test occurs when $cos(\ \widehat{Vec(U^i), Vec(U^j)}) > \Phi$ and as a result we want to bound the probability with which the above event may take place i.e.

$$Pr[cos(\widehat{Vec(U^i), Vec(U^j)}) > \Phi] \leq Pr[cos(\widehat{Vec(U^i), Vec(U^j)}) \geq \Phi] =$$

$$Pr[cos(\widehat{Vec(U^i), Vec(U^j)}) - cos(Vec(U^i), Vec(U^j)) \geq \Phi - cos(Vec(U^i), Vec(U^j))]$$

Moreover, since $\Phi \geq cos(Vec(U^i), Vec(U^j))$ the examined event may happen only when $cos(\widehat{Vec(U^i), Vec(U^j)}) > cos(Vec(U^i), Vec(U^j))$. As a consequence, we may equivalently consider the absolute values of the left and right side. Given this and expressing $cos(\widehat{Vec(U^i), Vec(U^j)})$ using its equivalent sum:

$$Pr[|cos(\widehat{Vec(U^i), Vec(U^j)}) - cos(Vec(U^i), Vec(U^j))| \geq |\Phi - cos(Vec(U^i), Vec(U^j))|] =$$

$$Pr[|\sum_{k=1}^{\omega} \frac{(\widehat{c_k^i \cdot c_k^j} - c_k^i \cdot c_k^j)}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|}| \geq |\Phi - cos(Vec(U^i), Vec(U^j))|] \leq$$

$$Pr[\sum_{k=1}^{\omega} \frac{|\widehat{c_k^i \cdot c_k^j} - c_k^i \cdot c_k^j|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} \geq |\Phi - cos(Vec(U^i), Vec(U^j))|]$$

The left side of the above inequality constitutes the sum of ω independent variables by recursively producing seeds for RHP vector generation as in the proof of Theorem 4. In the proof of Theorem 4, we showed that $\frac{|c_k^i \cdot c_k^j - \widehat{c_k^i \cdot c_k^j}|}{\|c_k^i\| \cdot \|c_k^j\|} \in 2|sin(\frac{\pi\epsilon}{2})|$ with probability $1-\delta$ and due to the fact that $\|c_k^i\| \cdot \|c_k^j\| \leq \|Vec(U^i)\| \cdot \|Vec(U^j)\|$ each such variable also lies in $0 \leq \frac{|\widehat{c_k^i \cdot c_k^j} - c_k^i \cdot c_k^j|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} \leq 2|sin(\frac{\pi\epsilon}{2})|$ with at least $1-\delta$ probability. Furthermore, due to the fact that the cosine receives values in $[-1, 1]$, in the worst case, with probability (at most) δ: $2|sin(\frac{\pi\epsilon}{2})| < \frac{|\widehat{c_k^i \cdot c_k^j} - c_k^i \cdot c_k^j|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|} \leq 2$.

Let $|X_k|$ denote the random variable that bounds the estimation error $\frac{|\widehat{c_k^i \cdot c_k^j} - c_k^i \cdot c_k^j|}{\|Vec(U^i)\| \cdot \|Vec(U^j)\|}$ for column k. $|X_k|$ has an exact (instead of at least) $1-\delta$ probability of being (not smaller or equal, but) simply equal to $2|sin(\frac{\pi}{2}\epsilon)|$ and 2 with probability δ. The expectation $E(e^{t \cdot X_k}) = (1-\delta) \cdot e^{t \cdot 2|sin(\frac{\pi\epsilon}{2})|} + \delta \cdot e^{2 \cdot t}$, for any $t > 0$. According to the generic form of the Chernoff bound [22] and our above discussion, due to the independence of X_ks (again, we emphasize that we assume independence of estimation errors based on the described manner of producing seeds during RHP application. Columns and their values are not assumed independent):

$$Pr[|cos(\widehat{Vec(U^i), Vec(U^j)}) - cos(Vec(U^i), Vec(U^j))| \geq |\Phi - cos(Vec(U^i), Vec(U^j))|] \leq$$

$$\min_{t>0} e^{-t \cdot |\Phi - cos(Vec(U^i), Vec(U^j))|} \prod_{k=1}^{\omega} E(e^{t \cdot X_k}) = \min_{t>0} e^{-t \cdot |\Phi - cos(Vec(U^i), Vec(U^j))|} \prod_{k=1}^{\omega} ((1-\delta) \cdot e^{t \cdot 2|sin(\frac{\pi\epsilon}{2})|} + \delta \cdot e^{2 \cdot t})$$

$$= \min_{t>0} e^{-t \cdot |\Phi - cos(Vec(U^i), Vec(U^j))|} ((1-\delta) \cdot e^{t \cdot 2|sin(\frac{\pi\epsilon}{2})|} + \delta \cdot e^{2 \cdot t})^{\omega}$$

which completes the proof. □

Bandwidth vs Accuracy Trade off. Based on Theorem 4, the tighter the quality guarantees of the framework become, i.e., both (ϵ, δ) decrease and thus $(|sin(\frac{\pi}{2}\epsilon)|, 1-\Delta)$ also decrease; the size of the bitmaps, proportional to $\ell log(2/\delta)/(2\epsilon^2)$, increases. This expresses how SWO-MuO trades bandwidth for similarity estimation accuracy. Regarding the similarity tests, Theorem 5 shows that, again, the probability of an incorrect similarity test decreases exponentially with $|\Phi - cos(Vec(U^i), Vec(U^j))|$.

SWO-MuO Trade off Practical Example. In case of picking a negligible δ, for instance $\delta = 0.001$, each $|X_k|$ will lie within $2|sin(\frac{\pi}{2}\epsilon)|$ almost surely and therefore the provided bound will approach $\min_{t>0} e^{-t \cdot |\Phi - cos(Vec(U^i), Vec(U^j))|} e^{t \cdot 2|sin(\frac{\pi\epsilon}{2})| \cdot \omega}$.

Now, for example, for δ as set above and $\epsilon = 0.05$ the proper bitmap size is only a fraction (e.g. 1/2, 1/4, 1/8, 1/16 etc) of *maximum* 83 bytes, which involves the consumed bandwidth for S_i per window slide. For instance, for $|\Phi - cos(Vec(U^i), Vec(U^j))| = 0.35$ and for a pair of recent column observations, the probability of an error in a similarity test is bounded by[1] $9.67 \cdot 10^{-31}$ for $t \rightarrow 1911.07$.

4.3. Sliding Window Operation - Unidimensional Outlier Detection (SWO-UnO) Operational Mode

We now elaborate on the fourth operational mode, SWO-UnO, which uses our unidimensional outlier definition and operates over sliding windows. Compared to SWO-MuO of the previous section, the difference now, because we focus on unidimensional outlier detection, is that clusterheads have to deduce correlations of individual rows with measurements in the current window. Recall that rows indeed correspond to samples of individual attributes (symbols summarized in Table 1) on which we focus upon identifying outliers for single dimensions. If we assume simple slides of the window of size equivalent to a single column c_ω each time then, per attribute, we can only compress a single measurement u^i_{rk} for each window slide which does not leave much space for data reduction.

Therefore, instead of simply doing that, we here examine more general sliding window cases where each slide in the window is performed in batches of columns [23]. That is, for every slide of the window, λ ($1 \leq \lambda \leq \omega$) columns expire and λ new columns are added as shown in Table 6. Note that for $\lambda = \omega$ this is equivalent to the TWO-UnO operational mode.

Slide ⇒					
10	12.2	11.1	12	⋯	18.3
21	⋱	⋱	⋱	⋮	28.6
⋮	⋮	⋮	⋮	⋱	⋮
31.9	32.1	31.1	32	⋯	38.3

Expired batch one per row		
10	12.2	11.1
21	⋱	⋱
⋮	⋮	⋮
31.9	32.1	31.1

New Window (batch ↓→)					
12	⋯	18.3	⟦8.75	8.8	9⟧
⋱	⋱	28.6	⟦4	5	3⟧
⋮	⋮	⋱	⋮	⋮	⋮
32	⋯	38.3	⟦31	29.9	30.1⟧

↑ new batch of each attribute (to compress)

Table 6: Exemplary exposition of the changes S_i's data undertake when operating in a sliding window paradigm of batches (in this example $\lambda = 3$).

4.3.1. SWO-UnO Processing at the Sensor Level

We assume all attribute values of the $r-$th attribute (row in Table 1) at mote S_i belonging to previous batch slides of the window have been compressed and transmitted to the clusterheads, so we concentrate on the local data processing the new batch $u^i_{rb_\omega} = [u^i_{r(\omega-\lambda)}, \cdots, u^i_{r\omega}]$ containing values of the $r-$th attribute ($1 \leq r \leq \xi$) should undertake. That is $u^i_{r(\omega-\lambda)}$ denotes the $\lambda-$th more recent measurement of the $r-$th attribute at mote S_i, placed inside the most recent batch that reaches the last value ω of the window $u^i_{rb_\omega}$ (Table 4). S_i applies the RHP LSH scheme on the vector $u^i_{rb_\omega}$ and derives a bitmap $X^i_{u_{rb_\omega}}$ of d length which is sent to the clusterhead along with $\|u^i_{rb_\omega}\|$. Let $B(\|u^i_{rb_\omega}\|)$ the size of the binary representation of the L_2 norm for the batch. Then, to achieve compression, we should ensure that $(d + B(\|u^i_{rb_\omega}\|))$ is (much) smaller than the size of the initial batch size λ. This can be tuned, as discussed in our subsequent analysis.

4.3.2. SWO-UnO Intracluster Processing

To explain our choices in the intracluster processing phase, we here show how the similarity estimation problem based on the compressed information in SWO-UnO can be transformed to an equivalent to the SWO-MuO operational mode. Note that the transformation we describe hereafter is just to show the aforementioned equivalence and does not need to be actually implemented by a clusterhead. Having said that, imagine we isolate the $r - th$ attribute from the U^i matrix in Table 1, which is instantiated and exemplified in Table 6. The corresponding isolated row $u^i_r \in R^\omega$ is a vector conceptually divided in a number of batches that are valid for the current window.

[1] Our formulas and calculations were validated using the Mathematica Software: Wolfram Research, Inc., Mathematica, Version 10.0, Champaign, IL (2014). In this particular case, the `FindMinimum` facility.

$$u_r^i = \Big[\ [\![32 \quad 35.3 \quad 38.8]\!] \quad [\![31.3 \quad 30.5 \quad 38.3]\!] \quad \cdots \quad [\![31 \quad 29.9 \quad 30.1]\!]\ \Big] \longrightarrow \begin{bmatrix} 32 & 31.3 & \cdots & 31 \\ 35.3 & 30.5 & \vdots & 29.9 \\ 38.8 & 38.3 & \cdots & 30.1 \end{bmatrix} = U_r^i$$

Table 7: Transforming the $r-$th attribute vector composed of batches ($\lambda = 3$) to a matrix where each column corresponds to a batch.

If we apply a transformation of the latter vector to a matrix U_r^i such that (also see Table 7): a) each column corresponds to a batch of λ size and thus there are λ rows in the newly formed matrix, b) the overall number of columns[2] in this matrix is $\frac{\omega}{\lambda}$, then we can see the resemblance of SWO-UnO **in a per attribute fashion** to how SWO-MuO operates on the whole U^i matrix during the intercluster processing phase. In particular:

- In both SWO-UnO and SWO-MuO in each slide of the window, the most aged column of U_r^i and U^i, respectively, expires and a new column is appended at the end of the window.
- At the sensor processing level, each mote compresses and transmitts the most recent table column in both operational modes.
- In the intra- and inter-cluster processing phases, clusterheads receive compressed represenations of individual columns $< X^i_{u_{rb_\omega}}, \|u^i_{rb_\omega}\| >$ for SWO-UnO and $< X^i_{c_\omega}, \|c^i_\omega\| >$ for SWO-MuO, but need to extract similarity estimations for the entire matrices U_r^i and U^i.
- The estimation of the partial inner product $\widehat{c^i_\omega \cdot c^j_\omega}$ extracted by a clusterhead in SWO-MuO (Equation 11) corresponds to $\widehat{u^i_{rb_\omega} \cdot u^j_{rb_\omega}}$ in the current SWO-UnO operational mode.

Given the above discussion, the receipt of $< X^i_{u_{rb_\omega}}, \|u^i_{rb_\omega}\| >$ of a mote S_i triggers clusterhead's C_i, where mote S_i is assigned to, effort in estimating $\widehat{u^i_{rb_\omega} \cdot u^j_{rb_\omega}}$. This is essential because the inner product of the overall cosine similarity of a pair of attributes is computed as (recall that according to our symbology, $u^j_{rb_\lambda}$ refers to the batch that reaches the $\lambda-$th measurement of the window, which is equivalent to the most aged batch):

$$u_r^i \cdot u_r^i = u^i_{rb_\lambda} \cdot u^j_{rb_\lambda} + \cdots + u^i_{rb_\omega} \cdot u^j_{rb_\omega} \tag{13}$$

and clusterhead C_i is able, for the most recent as well as for each previous slide, to derive an estimation of

$$\widehat{u^i_{rb_\omega} \cdot u^j_{rb_\omega}} \overset{Eq.\ 6}{=} cos\left(\frac{D_h(X^i_{u_{rb_\omega}}, X^j_{u_{rb_\omega}})}{d}\pi\right) \cdot \|u^i_{rb_\omega}\| \cdot \|u^j_{rb_\omega}\| \tag{14}$$

utilizing the received bitmaps. This is because $u^i_{rb_\omega} \cdot u^j_{rb_\omega} = cos(u^i_{rb_\omega}, u^j_{rb_\omega}) \cdot \|u^i_{rb_\omega}\| \cdot \|u^i_{rb_\omega}\|$ and $\|u^i_{rb_\omega}\|$,$\|u^j_{rb_\omega}\|$ are transmitted by motes, while $cos(u^i_{rb_\omega}, u^j_{rb_\omega})$ is approximated by $cos(\frac{D_h(X^i_{u_{rb_\omega}}, X^j_{u_{rb_\omega}})}{d}\pi)$. Having acquired such estimations for the parts of the inner product and because $\|u^i_r\| = \sqrt{\|u^i_{rb_\lambda}\|^2 + \cdots + \|u^i_{rb_\omega}\|^2}$, the overall similarity estimation will be:

$$\widehat{cos(u^i_r, u^j_r)} = \frac{\widehat{u^i_r \cdot u^j_r}}{\|u^i_r\| \cdot \|u^j_r\|} \tag{15}$$

Based on the estimation of Equation 15, similarity tests $\widehat{cos(u^i_r, u^j_r)} \geq \phi_r$ are performed and the motes that are found similar gain support for their $r-$th attribute, i.e., $Sup^i_r \leftarrow Sup^i_r + 1$ and $Sup^j_r \leftarrow Sup^j_r + 1$. Finally, motes that do not reach the required $minSup_r$ are added to the $PotOutC_i$ list.

[2] We assume ω mod $\lambda = 0$ to simply showcase the transformation.

4.3.3. SWO-UnO Intercluster Processing

The intercluster processing phase takes place only for those motes that did not manage to accumulate $\geq minSup_r$ support for their $r-$th attribute. The information that $PotOut_{C_i}$ transmits, includes information per attribute, for each attribute of mote S_i for which $Sup_r^i < minSup_r$. If S_i is placed in $PotOut_{C_i}$, for its $r-$th attribute, after $1 \leq \tau_r < \frac{\omega}{\lambda}$ shifts of the window, C_i needs to form a triplet of $< S_i, \{< X^i_{u_{rb_{(\tau_r \cdot \lambda)}}}, \|u^i_{rb_{(\tau_r \cdot \lambda)}}\| >, \cdots, < X^i_{u_{rb_\omega}}, \|u^i_{rb_\omega}\| >\}, Sup_r^i >$ to transmit to its TSP neighbor. For clarity, we again stress that according to our notation $< X^i_{u_{rb_{(\tau_r \cdot \lambda)}}}, \|u^i_{rb_{(\tau_r \cdot \lambda)}}\| >$ will include the respective bitmap and norm value for the $u^i_{rb_{(\tau_r \cdot \lambda)}}$ batch, which in turn involves $[u^i_{r(\tau_r \cdot \lambda - \lambda)}, \cdots, u^i_{r(\tau_r \cdot \lambda)}]$ attribute values.

As in the SWO-MuO case, the value of τ_r may differ for each clusterhead pair on a respective TSP edge. This is because if data about a mote S_i reaches a clusterhead where its $r-$th attribute finds adequate support, the intercluster communication ends for that (mote, attribute) pair. Since each cluster knows the number of window slides since the last intercluster communication with its TSP neighbor, it can compute τ_r for each attribute case and communicate only the part of $\{< X^i_{u_{rb_\lambda}}, \|u^i_{rb_\lambda}\| >, \cdots, < X^i_{u_{rb_\omega}}, \|u^i_{rb_\omega}\| >\}$ that is missing from its TSP neighbor for the current window.

4.3.4. SWO-UnO Analysis

The following lemmas regarding the estimation accuracy of SWO-UnO (Lemma 3) and its accuracy with respect to the similarity tests (Lemma 4), evolve from Theorem 4 and Theorem 5 based on our discussion in this section.

Lemma 3. *For bitmaps of $O(\ell log(2/\delta)/(2\epsilon^2))$ length (Theorem 2) the estimation of Equation 15 lies in:*

$$\widehat{cos(u_r^i, u_r^j)} \in cos(u_r^i, u_r^j) \pm 2sin(\frac{\pi}{2}\epsilon)$$

with probability at least $\Delta = (1-\delta)^{\frac{\omega}{\lambda}}$.

Lemma 4. *For bitmaps of $O(\ell log(2/\delta)/(2\epsilon^2))$ length (Theorem 2), clusterheads perform a correct similarity test upon utilizing $\widehat{cos(u_r^i, u_r^j)}$ (Equation 15) with probability at least $1 - \min_{t>0} e^{-t \cdot |\phi_r - cos(u_r^i, u_r^j)|}((1-\delta) \cdot e^{t \cdot 2|sin(\frac{\pi\epsilon}{2})|} + \delta \cdot e^{2 \cdot t})^{\frac{\omega}{\lambda}}$.*

With respect to the Bandwidth vs Accuracy Trade off, similar observations as in the SWO-MuO case can be extracted for the SWO-UnO case, as well.

5. Related Work

The focal point of our work, i.e., outlier detection in sensor networks has received considerable attention over the past few years [7, 24, 25]. The same holds for Locality Sensitive Hashing [26] which is enhanced in this work, so that it becomes applicable to the examined operational modes during outlier detection. The works in [7] and [26] present comprehensive surveys on outlier detection in sensor networks and LSH, respectively. Here, we comment on works that we consider more relevant to our proposed approaches and representative of wider categories due to the rationale they employ.

The TACO framework that was the first outlier detection technique that could provide tunable accuracy guarantees based on bandwidth constraints, was initially introduced in [4]. Moreover, [5] includes enhancements in the aspects of the adoption of various similarity measures within the framework and the probabilistic guarantees provided by TACO's techniques. This work significantly extends TACO's paradigm with respect to the aspects extensively discussed upon citing our contributions in the introductory section. The Locality Sensitive Hashing (LSH) scheme used in TACO is discussed in [10], while [27] provides interesting data sensitive extensions to the basic LSH technique. The techniques we present in this work employ the same LSH scheme, but in order to enable its usage in a sliding window paradigm and during multidimensional outlier detection procedures, we develop a thorough study on required data transformations. Furthermore, we present an extensive analysis on the probabilistic guarantees of our framework's accuracy entailed in the more generic operational modes that we introduce.

The work of [3] uses an equivalent to ours outlier definition and pinpoints outlier motes utilizing a geometric approach in order to allow sensor nodes avoid any communication unless their similarity is possible to have crossed a given threshold. Nonetheless, the main focus of the method is on pair-wise mote comparisons and its quality

guarantees may not hold when it comes to handle minimum support based outlier definitions in hierarchical sensor networks. The PAO framework [6] operates on top of clustered network organizations, as our work does, and restrains communication during outlier detection by employing linear-regression based compression. Nonetheless, the amount of compression and thus the communication savings strongly depend on the ability of the monitored quantities to exhibit low fluctuations in order to enable linear regression's application. The authors of [28] introduce a declarative data cleaning mechanism over data streams produced by the sensors. Similarly, the work of [29] introduces a data cleaning module designed to capture noise in sensor streaming data based on the prior data distribution and a given error model $N(0, \sigma^2)$. A different approach is presented in [30], where Pairwise Markov Networks are used as a tool to derive a subset of motes sufficient to infer the values obtained by the whole network. However, this technique requires an energy draining learning phase. In [31] a probabilistic technique for cleaning RFID data streams is presented. The framework of [2] is used to identify and remove outliers during the computation of aggregate and group-by queries. In [8] the authors introduce a novel definition of an outlier, as an observation that is sufficiently far from most other observations in the data set. However, in cases where the motes observe physical quantities (such as noise levels, temperature) the absolute values of the acquired readings depend, for example, on the distance of the mote from the cause of the monitored event (i.e., a passing car or a fire respectively). Thus, correlations among readings in space and time are more important than the absolute values, used in [8].

The work in [32, 33] addresses the problem of identifying faulty sensors using a localized voting protocol. However, localized voting schemes are prone to errors when motes that observe interesting events generating outlier readings are not in direct communication [2]. In other related work, [34] proposes a fuzzy approach to infer the correlation among readings from different sensors, assigns a confidence value to each of them, and then performs a fused weighted average scheme.

As already noted the current work differentiates itself by providing a straightforward manner to trade bandwidth consumption for accuracy during the outlier detection procedure and, remarkably, it does so in four operational modes.

6. Experimental Evaluation

6.1. Experimental Set up

We employ the same enhanced, real-world datasets as in [4, 5, 35]. More precisely the characteristics of these datasets are as follows:

- Intel Lab Data: Measurements for 4 monitored attributes including temperature, relative humidity, light and voltage obtained by 48 sensors in the Intel Research, Berkeley lab[3].
- Weather Data: Measurements for 6 monitored attributes of 100 sensor nodes including air temperature, dew-point temperature, wind speed, wind peak, solar radiance and relative humidity for the station in the university of Washington, and for year 2002 [4].

For TWO-MuO and SWO-MuO, which involve multidimensional outlier detection, we normalize the dataset in a per attribute fashion, subtracting the mean value from each measurement. This is to avoid an extreme number of detected multidimensional outliers being affected by the heterogeneity of the absolute attribute values in the datasets. However, for SWO-UnO where unidimensional outlier detection is performed, this issue does not arise and thus we consider the original measurements. Each mote measurement corresponds to a real valued attribute of 4 bytes=32 bits size.

We measure the accuracy (approximation quality) of our techniques in detecting outliers using the $F-measure = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$ metric. In a nutshell, precision specifies the percentage of reported outliers that are true outliers, while recall specifies the percentage of outliers that are reported. The set of true outliers was computed offline (i.e. assuming all data was locally available) in each experiment. Our omnibus solutions pinpoint outliers in an approximate way, trading bandwidth for accuracy in a straightforward manner. We reduce the size of the transmitted data leveraging

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

[4] http://www-k12.atmos.washington.edu/k12/grayskies

data reduction capabilities of the windowed LSH techniques we propose separately for TWO-MuO, SWO-MuO and SWO-UnO operational modes. The more we reduce the bitmap size, the looser the guarantees on the accuracy of the similarity tests we conduct, as discussed in our analysis in the previous sections.

According to the specifications in Section 4.1, in TWO-MuO sensor nodes collect ω samples for the ξ monitored attributes, where ξ =4 for Intel and ξ =6 for the Weather data (also see Table 1). Based on these, sensors perform the intra- and intercluster communication phase and, due to the tumbled window adoption, these $\xi \times \omega$ measurements expire altogether for the next tumble to be filled and processed. Therefore, because each real-valued measurement is of 32 bits size, the size of the original data is $\xi \cdot \omega \cdot 32$ bits. These are the data sent by the *NoReduction* approach mentioned in our experiments upon operating in TWO-MuO mode. Upon we impose data reduction as described in Section 4.1 employing $1/x$ *Reduction* ratio, $\frac{\xi \cdot \omega \cdot 32}{x}$ bits are communicated during intra- and intercluster (for sensors that do not obtain adequate support) phases. This is denoted by $1/x$ *Reduction* ratio in our evaluation.

For SWO-MuO where a sliding window is used, in every slide of the window the most aged measurements expire while sensor nodes collect new ξ measurements composing the latest information, c_ω column in Table 1, that need to be transmitted during the intra- and intercluster communication phases. Further details were discussed in Section 4.2. Therefore, in this case the *NoReduction* approach sends the original $\xi \cdot 32$ bits sized data in every slide of the window. Upon we impose $1/x$ *Reduction* ratio as described in Section 4.2, $\frac{\xi \cdot 32}{x}$ + 32 bits are communicated during the intra- and intercluster (in accordance with our discussion in Section 4.2) phases. The addition of 32 bits in the previous calculation is due to the fact that SWO-MuO also communicates the L_2 norm of the c_ω column.

Finally, for SWO-UnO (Section 4.3) a slide in the window causes, for each monitored attribute, the expiration of λ measurements in the most aged batch and the addition of a new batch, also composed of λ measurements. Therefore, in every slide of the window and for each monitored attribute the information that is communicated by *NoReduction* is of size: $\lambda \cdot 32$ bits while $\frac{\lambda \cdot 32}{x}$ + 32 bits are transmitted for $1/x$ *Reduction*.

It is important to emphasize that in each operational mode, the *NoReduction* approach means that we transmit the original sensor data. Therefore, the accuracy of the outlier detection process is perfect for the *NoReduction* approach. Hence, in our accuracy related plots we do not include a separate line for *NoReduction* since it would simply be a line parallel to the horizontal axis at $F - measure = 1$ (vertical axis). We do include *NoReduction* in communication and energy performance (details follow) graphs.

In our tests we vary every possible parameter that can alter accuracy, communication and energy consumption in our experiments over both datasets. Each experiment is repeated 10 times under identical parameterizations altering only the random vectors used in the windowed LSH application per operational mode. We then report the average over these 10 repetitions. Parameterization, on the other hand, involves the following: (a) the reduction ratio imposed by the application, as discussed above, (b) the window size ω, (c) the $minSup$ parameter, (d) the similarity threshold Φ_θ (for TWO-MuO) or Φ (for SWO-MuO, SWO-UnO), (e) the number of attributes by using Intel data with $\xi = 4$ and Weather data with $\xi = 6$, (f) the batch size λ for SWO-UnO. We emphasize that we choose to directly tune the reduction ratio in parameter (a) instead of parameters ϵ, δ separately because, by doing so, we actually account for different combinations of ϵ, δ in the construction of the bitmaps as well as it is much more intuitive from an application viewpoint. For instance, reducing by 1/4 a bitmap of 1024 bits, corresponds to choosing any (δ, ϵ) pair as follows: $2 \cdot 10^{-512} < \delta < 1, \epsilon = 0.029 \cdot \sqrt{\ell og(2/\delta)}$.

We virtually place the sensors of each dataset in a square 100×100 meters area by having each mote choose a random location for each set of tested parameters. We organize the network in four clusters (and thus clusterheads) of $ClusterSize = 12$ and $ClusterSize = 25$ for the Intel and Weather dataset, respectively. The maximum packet size of communication, equivalent to TOSH_DATA_LENGTH in [36], is set to 32 bytes=256 bits. The energy consumption while transmitting and receiving data is modeled according to [37]. In particular, transmitting b bits of data to a node that lies at a distance $dist$ from the current node results in an energy drain of $(E_{TX} + E_{RF} \times dist^2) \times b$, where E_{TX} denotes the per bit power dissipation of the transmitter electronics and E_{RF} denotes the per bit and squared distance power delivered by the power amplifier. Similarly, receiving b bits of data results in an energy drain of: $E_{RX} \times b$. The values of these parameters are set similarly to [37] as: $E_{TX} = E_{RX} = 50nJ/bit$ and $E_{RF} = 100pJ/bit/m^2$.

We also model potential communication channel losses. In our experiments, the transmitted messages have a probability of requiring a retransmission due to message loss or collision. We use a (low, for instance in [16] losses can reach > 50% of the total messages) $P_{loss} = 0.01$ probability of a message loss, while the probability of a message collision is modeled as in [38] with $P_{collision} = 1 - e^{-2 \cdot \frac{m \cdot T_p}{T - T_p}}$. In our simulation we assume that sensor nodes should

report their latest measurements according to the tested operational mode within $T = 1$ sec at the end of each new window. Furthermore, m denotes the number of the sensor nodes each of which picks uniformly at random a slot from the T sized time frame to transmit its data. For intracluster processing m is equal to the cluster size for each dataset, while $m = 4$ during the intercluster communication phase i.e., equal to the number of clusterheads. Finally, $T_p = \frac{transmitted\ data\ size}{bandwidth}$ where for *bandwidth* we use the 38.4 Kbps of Mica2 mote[5]. Based on the above, apart from accounting for accuracy for each operational mode and utilized dataset, our experiments also present results on the amount of communicated data and consumed energy throughout the outlier detection process.

6.2. Summary of Main Findings

Before proceeding into the detailed analysis of our results, we here highlight the main findings extracted throughout the evaluation process:

- In accordance with the theoretic results of our work, the more we compress the data choosing a high x in the $1/x$ *Reduction* ratio, the lower the accuracy (F-Measure) can become. However, we found out that doubling or even quadrupling x, i.e., from 1/4 to 1/8 and 1/16 causes only slight deterioration of accuracy in the vast majority of the cases. This means that we can maximize communication savings by reducing the data size without introducing significant (additional) inaccuracies.

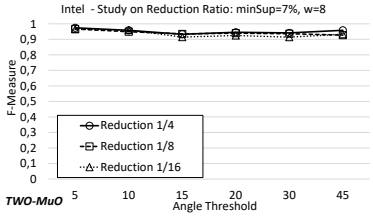
- In the presence of message collisions and retransmissions our techniques exceed their expected theoretic benefits in terms of bandwidth preservation and energy consumption. For instance, imposing $1/16$ *Reduction* ratio in TWO-MuO can protract the expected network lifetime by 21, instead of expectedly 16, times.

- According to Theorem 4 (SWO-MuO) and Lemma 3 (SWO-UnO) the larger the ω we choose the smaller (closer to 1) the Δ probability on the approximation error becomes. Therefore, one would expect that upon increasing ω in SWO-MuO or SWO-UnO the accuracy of our techniques would deteriorate. However, in practice this rarely happens, due to the fact that the more observations we include in the window, the more distinguishable sensors producing outlying measurements become.

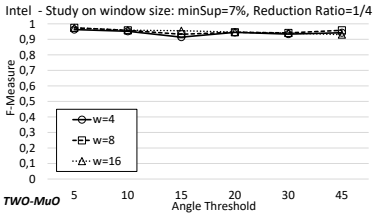
- Our framework is expectedly more accurate with stricter Φ_θ (for TWO-MuO) or Φ (for SWO-MuO, SWO-UnO) thresholds. This was explicitly mentioned in Section 4.1 for TWO-MuO, but similar observations can be extracted for the exponential bounds in Theorem 5 (SWO-MuO) and Lemma 4 (SWO-UnO). Nonetheless, in practice we found out that, contrary to TWO-MuO which operates on tumbles, SWO-MuO and SWO-UnO are more affected by lower Φ values and can overall exhibit a lower $F - measure$ score. The reason for this is that: each random vector used in the windowed RHP-LSH application (irrespectively of the operational mode) draws a hyperplane that partitions the data space in two sides corresponding to bit 0 or 1 for each position of the bitmaps. There might be a case where the produced random vectors do not ideally partition the space so that outlying data points can be distinguished using the bitmaps. Tumbled window operation restricts by definition the effect of these "bad" random vectors in a single window since afterwards all results expire and a new window is formed. In sliding window operation, however, compressed representations are reused in the similarity tests, ω and ω/λ times until the corresponding column for SWO-MuO or batch for SWO-UnO expires. Thus, "bad" random vectors affect multiple times the accuracy of the results. This issue can be overcome by using derived dimensions for choosing random vectors as in [27] or more specialized LSH alterations [26]. However, applying these approaches on our omnibus solutions is out of the scope of the current work since it essentially involves comparing orthogonal LSH versions.

- In TWO-MuO we gain more in terms of bandwidth and energy consumption in all the cited cases. This is because in TWO-MuO (a) we compress higher amounts of data (as already mentioned of $\xi \cdot \omega \cdot 32$ bits) compared to SWO-MuO ($\xi \cdot 32$ bits) and SWO-UnO ($\lambda \cdot 32$), so in absolute numbers $1/x$ *Reduction* has a stronger effect on the bitmaps, (b) we do not need to add 32 bits in every transmission since TWO-MuO does not require any additional L_2 norm to be encapsulated in a message.

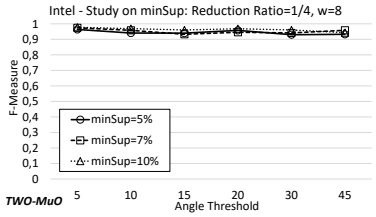
[5] http://www.willow.co.uk/MoteWorks_OEM_Edition.pdf



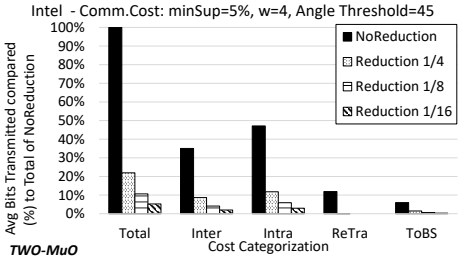
(a) TWO-MuO Intel: Reduction Ratio



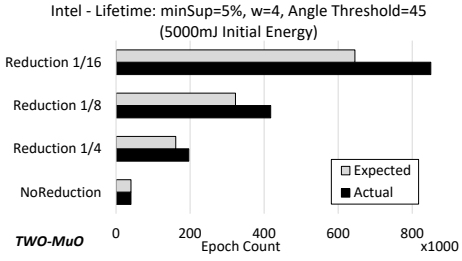
(b) TWO-MuO Intel: Window Size



(c) TWO-MuO Intel: minSup

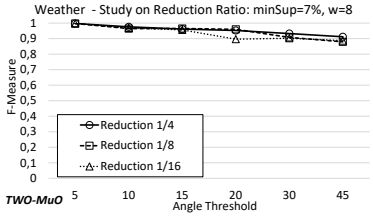


(d) TWO-MuO Intel: Communication Cost

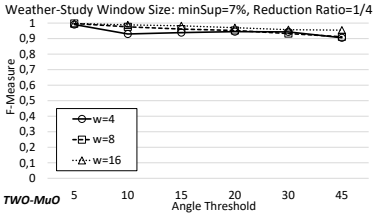


(e) TWO-MuO Intel: Energy Consumption

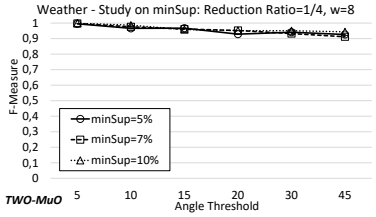
Figure 3: TWO-MuO Analysis on Intel Data



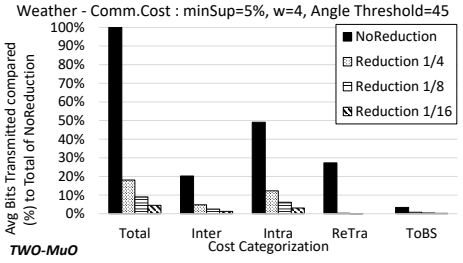
(a) TWO-MuO Weather: Reduction Ratio



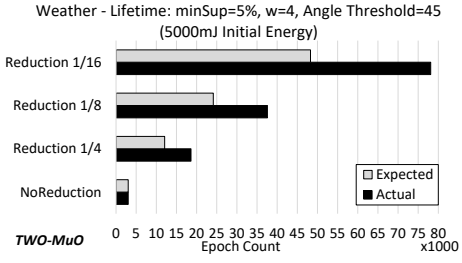
(b) TWO-MuO Weather: Window Size



(c) TWO-MuO Weather: minSup



(d) TWO-MuO Weather: Communication Cost



(e) TWO-MuO Weather: Energy Consumption

Figure 4: TWO-MuO Analysis on Weather Data

6.3. TWO-MuO Experimental Analysis

Figures 3(a) and Figure 4(a) present the accuracy of TWO-MuO, for the Intel and Weather data, for different angle (and thus cosine) similarity thresholds between 5 and 45 degrees and for different reduction ratios. We extract these results using a window size of $\omega = 8$ and $minSup = 7\%$, but shortly we are going to show how we vary the latter parameters as well. In Figures 3(a), 4(a) we observe that in both datasets we can impose a high degree of compression from 1/4 to 1/16 Reduction while keeping the accuracy of TWO-MuO above 90% in all the cited cases. In fact, in both datasets doubling or even quadrupling the reduction ratio has only a slight effect (no more than 5%) on F-Measure. On the other hand, the value of F-Measure slightly decreases, especially in Figure 4(a), as we increase

the angle similarity threshold on the horizontal axes, which is the expected behavior according to our discussion in Section 4.1 and Figure 2.

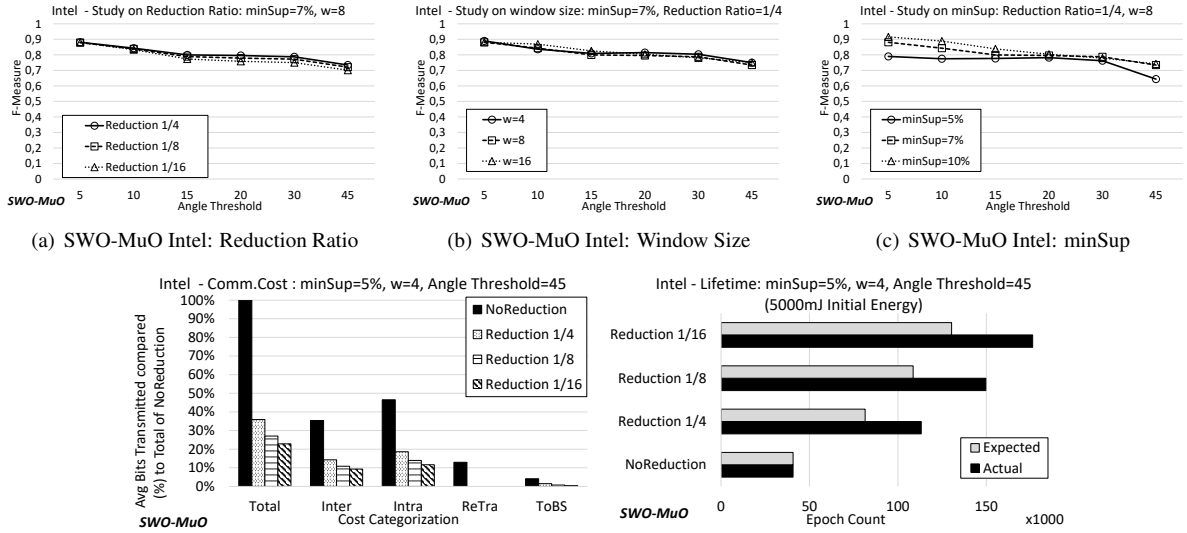
In Figures 3(b), 4(b) we keep a steady reduction ratio of $1/4$ (as shown above the reduction ratio does not tremendously affect accuracy) and $minSup = 7\%$ varying the window size from $\omega = 4$ to $\omega = 16$ sampled $\xi-$dimensional (attribute) values under different similarity thresholds. Again, the accuracy of TWO-MuO shows no remarkable fluctuation among the window choices. This is also true in Figures 3(c), 4(c) where for $1/4$ reduced bitmaps and $\omega = 8$, the accuracy of TWO-MuO remains resilient to varying $minSup$ in the range between 5% and 10% of the total sensor nodes in a dataset.

For measuring the performance of TWO-MuO in terms of bandwidth usage and energy consumption we pick a high similarity threshold of 45 degrees angle, the smallest of the tested window sizes $\omega = 4$ and a $minSup = 5\%$. The reason for these choices is to show the gains of TWO-MuO compared to *NoReduction* in a scenario that significantly favors *NoReduction*. The aforementioned scenario favors *NoReduction* because *NoReduction* transmits larger or even multiple messages when (uncompressed) data do not fit in a single packet, i.e., for higher ω choices, and therefore, it is more prone to message losses, collisions and retransmissions. In conjunction with the previous fact, upon loosening the outlier definition using high threshold and low $minSup$, the amount of data that *NoReduction* communicates is less since outliers become rarer.

Given these, in Figures 3(d) and 4(d) we can see the $Total$ bits transmitted by TWO-MuO versus the *NoReduction* approach. Going one step further, we also provide the categorization of communication during to the *Intra*-cluster communication phase, the *Inter*-cluster phase, *Retra*-nsmissions and $ToBS$ for final outliers reported to the BaseStation in the Intel and Weather data. To keep the results comparable among different datasets and operational modes (in the upcoming experiments), instead of reporting the absolute number of transmitted bits, we report the amount of communication of each category and $1/x$ *Reduction* ratio as a percentage of the bits required in $Total$ by the *NoReduction* approach. Thus, the $Total$ of *NoReduction*, black filled bar on the left of the figures, corresponds to 100% and the rest of the percentages express the ratio over this value.

In both Figure 3(d) and Figure 4(d), from the corresponding bars in the $Total$ category, we can observe that using TWO-MuO we gain more than the chosen reduction ratio yields. For instance, using $1/8$ *Reduction* in Intel data results in reducing the transmitted bits by a factor of 9.5 instead of 8, while this factor reaches a value of 11 in Weather data. These gains are attributed to two reasons: 1) the amount of retransmissions in the $ReTra$ category which are almost absent from our TWO-MuO, $1/x$ *Reduction* ratios, but reach 12% in Intel and 23% in Weather data for the *NoReduction* approach. Note that this difference is because for the same $\omega = 4$, Weather data are composed of more attributes and thus transmitted messages are larger, b) the amount of messages in the *Inter*-cluster category, where we have multiple (up to 4) hops of communication and thus, especially in the Intel data, compressed versus uncompressed communication contributes multiple times. It is worthwhile noticing that the percentage of *Inter* is in all cases less than the corresponding percentage of *Intra* in both datasets. This shows that the intercluster communication phase actually helps in reducing the overall communication in the network. This is because the naive approach of centralizing all the data (often called $SelectStar$ approach [4, 5]), after the intracluster communication phase, would have a cost equivalent to $2 \cdot Intra$ of the *NoReduction* approach. Now, in both figures $Intra + Inter < 2 \cdot Intra$ and therefore our valid claim.

In Figures 3(e) and 4(e) we depict the network lifetime for TWO-MuO under $1/x$ *Reduction* compared to the *NoReduction* approach for sensor nodes that are initialized with 5000 mJ residual energy. Throughout our evaluation, the network lifetime is defined as the epoch (corresponding to the periodic timepoints at which ξ monitored attributes are sensed by the motes) when the first mote in the network completely depletes its energy. For instance, upon using a $1/8$ *Reduction* ratio, Figure 3(e) shows that we can protract the network lifetime by 10,4 times. Note that this is more than the 9.5 factor in terms of transmitted bits that was reported in 3(d). Similarly, for a $1/8$ *Reduction* ratio, Figure 4(e) shows that the network lifetime is prolonged by 12.5 times, despite that 11 times fewer transmitted bits were observed in Figure 4(d). The reason for this difference among communication and energy consumption reduction is that measuring transmitted bits charges only the sensors that perform the transmission, while measuring the consumed energy, apart from charging the transmitting motes, also charges receiving nodes $E_{RX} \times b$ as discussed in Section 6.1. This is especially important for the bits of the $ReTra$ category, since TWO-MuO under any $1/x$ *Reduction* ratio almost completely avoids retransmissions. Notice that in both Figures 3(e) and 4(e) the $Expected$ network lifetime based on the imposed reduction ratio is plotted under the gray bar, and it is always exceeded by the black



(a) SWO-MuO Intel: Reduction Ratio (b) SWO-MuO Intel: Window Size (c) SWO-MuO Intel: minSup

(d) SWO-MuO Intel: Communication Cost (e) SWO-MuO Intel: Energy Consumption

Figure 5: SWO-MuO Analysis on Intel Data

bar of the *Actual* lifetime prolongation. The higher network lifetime prolongation, provided by TWO-MuO, reaches a factor of 21, in Weather data for $1/16$ *Reduction*.

6.4. SWO-MuO Experimental Analysis

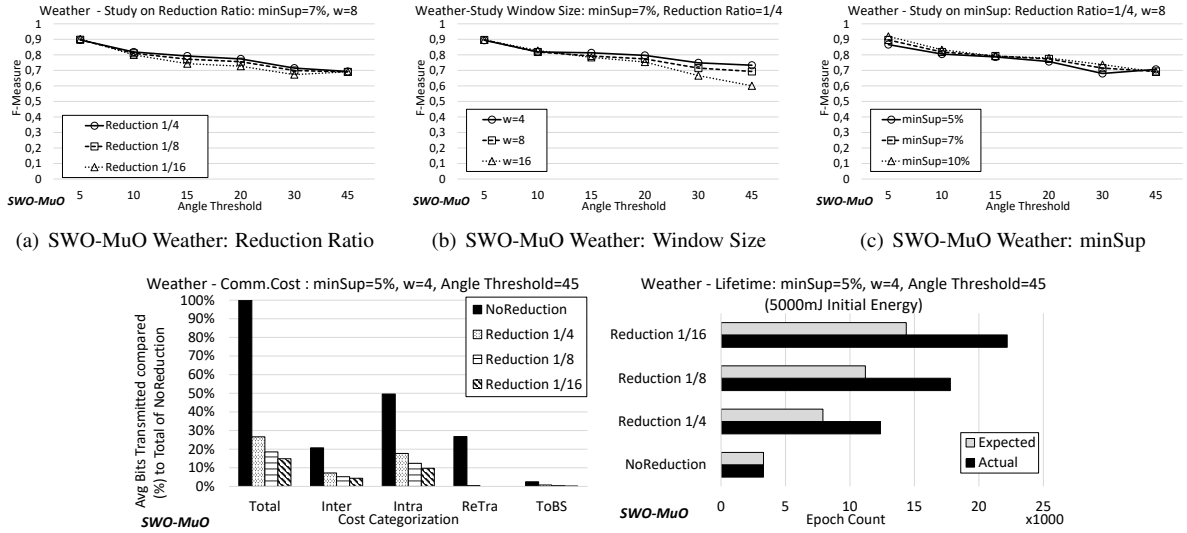
Figures 5(a) and Figure 6(a) present the accuracy of SWO-MuO, for the Intel and Weather data, for different angle (and thus cosine) similarity thresholds and reduction ratios. In both datasets the value of the F-Measure metric ranges between 70% and 90% mostly revolving around 80%. Once again it appears that we can impose a high degree of compression from 1/4 to 1/16 Reduction while not affecting the accuracy of SWO-MuO. In fact, in both datasets, doubling or even quadrupling the reduction ratio may affect F-Measure no more than 4%.

The value of F-Measure more steeply (compared to TWO-MuO) decreases as we increase the angle (equivalently decrease the cosine) similarity threshold, which is the expected behavior as we discussed in Section 6.2 and stems from Theorem 5. Overall, in all cited cases we can see that the F-Measure metric for SWO-MuO is lower from 5% (for 5 degrees) to 25% (for 45 degrees) compared to the F-Measure metric of TWO-MuO. This is because occasional "bad" random vectors that are incapable of properly partition (via the drawn half-planes) the monitored data space affect the results of the conducted similarity tests ω times until the corresponding attribute column (c_ω in Table 1) expires, contrary to TWO-MuO where such random vectors are not only rare but also affect one tumble. This important difference was also mentioned in the highlights of Section 6.2.

According to Theorem 4, the probability Δ on the approximation quality of SWO-MuO becomes smaller as ω increases. Nevertheless, Figure 5(b) illustrates that in practice increasing the window size from $\omega = 4$ to $\omega = 16$ for Intel data does not have an important impact on the SWO-MuO's F-Measure. This is also true for the Weather data in Figure 6(b) where the only important (10-13% decrease in F-Measure) decrement arises in 30-45 degrees for $\omega = 16$.

Varying the $minSup$ parameter from 5% to 10% in the Intel data can deteriorate the F-Measure value up to 10% with SWO-MuO being more accurate for higher $minSup$ as shown in Figure 5(c). On the other hand, this is does not appear to be the case for the Weather data since in Figure 6(c) altering $minSup$ causes no significant change in F-Measure. To further investigate this behavior we conducted experiments using synthetic datasets (motes were sampling values from a standard Gaussian distribution increasing the number of dimensions from 4 to 12). We omit the corresponding plots, but the result can be summarized to the fact that the higher the dimensionality (ξ monitored attributes) of the data, the more important the effect of $minSup$, with higher $minSup$ yielding higher F-Measure.

With respect to bandwidth consumption both Figure 5(d) and Figure 6(d) show, in the corresponding bars in the $Total$ category, that using SWO-MuO we gain more than the chosen reduction ratio yields. For instance, using



(a) SWO-MuO Weather: Reduction Ratio (b) SWO-MuO Weather: Window Size (c) SWO-MuO Weather: minSup

(d) SWO-MuO Weather: Communication Cost (e) SWO-MuO Weather: Energy Consumption

Figure 6: SWO-MuO Analysis on Weather Data

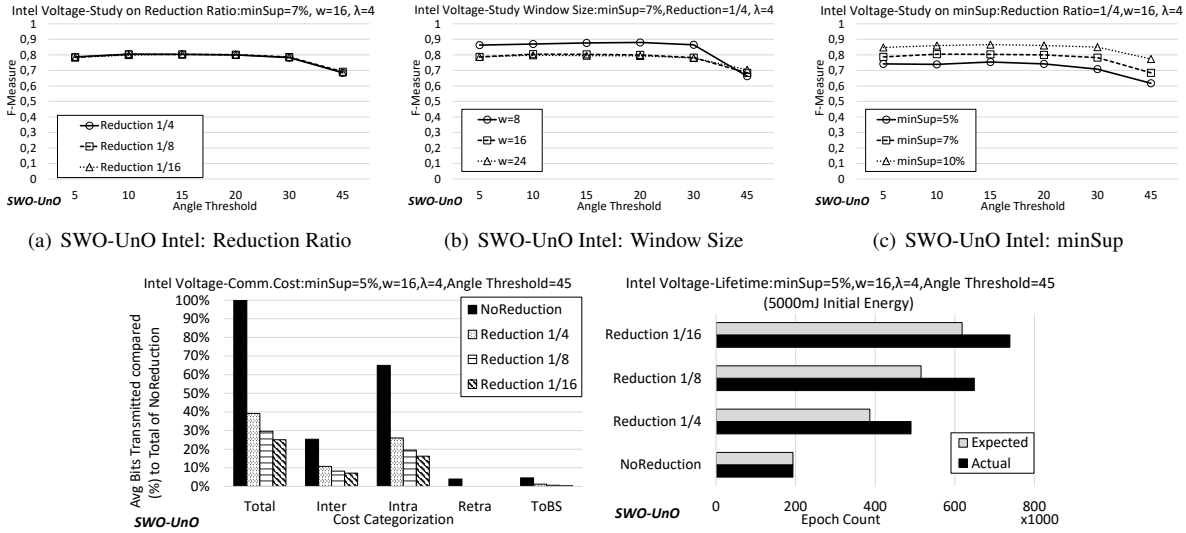
$1/8$ *Reduction* in Intel data results in reducing the transmitted bits by a factor of 3.6 instead of 2.67 (because SWO-MuO $1/4$ *Reduction* transmits $\frac{\xi \cdot 32}{x} + 32$ with $\xi = 4$ for Intel – see Section 6.1), while this factor reaches a value of 5.4 in Weather ($\xi = 6$) data instead of the expected 3.43. These gains are again attributed to the amount of retransmissions in the *ReTra* category that aggravate especially the *NoReduction* approach, as well as the amount of messages in the *Inter*-cluster category. In all the cited cases and also for the rest of the reduction ratios, SWO-MuO exceeds its theoretic benefits. Once again, the fact that $Intra + Inter < 2 \cdot Intra$ in all cases illustrates that using the intercluster communication framework saves communication compared to the naive approach that centralizes all the data to the base station after the intracluster communication phase.

It can easily be observed that, under the same reduction ratios, the bandwidth preservation over the *NoReduction* approaches is much higher for TWO-MuO than for SWO-MuO, both on expectation and in practice. This comes as no surprise according to our discussion in Section 6.1. For instance, for $\omega = 4$ and $\xi = 4$ under $1/8$ *Reduction*, TWO-MuO yields $\frac{4 \cdot 4 \cdot 32}{8}$ sized bitmaps compared to $4 \cdot 4 \cdot 32$ of *NoReduction*. On the other hand, for SWO-MuO $1/8$ *Reduction* yields $\frac{4 \cdot 32}{8} + 32$ sized bitmaps compared to $4 \cdot 32$ bits of *NoReduction*. As already discussed, $1/x$ *Reduction* has a stronger effect in TWO-MuO which stems from the definition of tumbled and sliding window operation.

In Figures 5(e) and 6(e) we depict the network lifetime for SWO-MuO under $1/x$ *Reduction* compared to the *NoReduction* approach for sensor nodes that are initialized with 5000 mJ residual energy. Again, in all cases SWO-MuO exceeds its theoretic (according to the imposed reduction ratio) benefits. For instance, upon using a $1/8$ *Reduction* ratio, Figure 5(e) shows that we can protract the network lifetime by 3.7 times instead of the expected (gray bar) 2.67 factor. Similarly, for a $1/4$ *Reduction* ratio, Figure 6(e) shows that the network lifetime is prolonged by 5.5 times instead of the expected 3.43 factor. Notice that in both Figures 5(e) and 6(e) the *Actual* network lifetime always exceeds the *Expected* lifetime prolongation.

6.5. SWO-UnO Experimental Analysis

SWO-UnO operates on individual attributes were in each slide of the window and a per attribute fashion, it compresses batches of measurements. Among the available attributes we choose to report results about the most interesting cases of a slow to change and narrow range ($[0, 4]$) attribute, namely the Intel Voltage attribute and the extreme case of Weather Solar measurements that have a large range among 0 and 1000 and exhibit abrupt changes depending on factors such as clouds, shades of objects etc. For the Intel Voltage measurements we choose a batch size $\lambda = 4$, while this value is set to 8 for the Weather Solar data. For these attributes we use window sizes that range between $\omega = 8$ (minimum in Intel Voltage) and $\omega = 48$ (maximum in Weather Solar) measurements so that



(a) SWO-UnO Intel: Reduction Ratio (b) SWO-UnO Intel: Window Size (c) SWO-UnO Intel: minSup

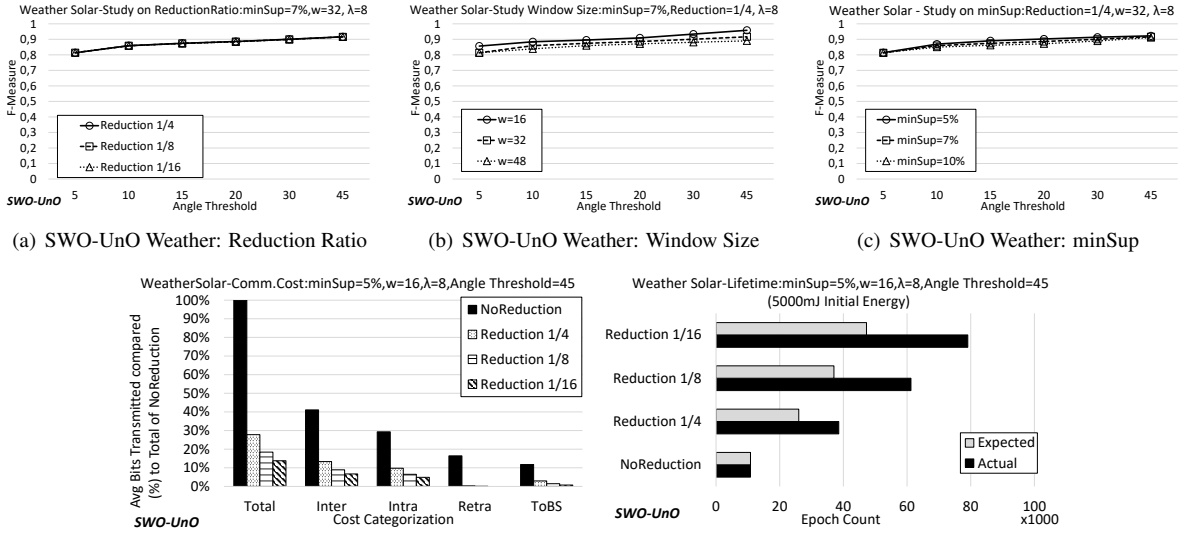
(d) SWO-UnO Intel: Communication Cost (e) SWO-UnO Intel: Energy Consumption

Figure 7: SWO-UnO Analysis on Intel Data

they contain a sufficient number of batches and $\frac{\omega}{\lambda}$ (Lemma 4) remains the same for cases examined in both datasets. Furthermore, we note that we report results for each attribute separately as if outliers are monitored per attribute in isolation. This gives us the opportunity to better conceive the pure effect of SWO-UnO per attribute. In case of examining unidimensional outliers but simultaneous monitoring for all attributes, the communication and energy performance of SWO-UnO would lead to observations similar to those of TWO-UnO.

Figure 7(a) and Figure 8(a) present the accuracy of SWO-UnO, for the Intel and Weather data, for different reduction ratios. In the Intel Voltage data (Figure 7(a)) the value of the F-Measure metric steadily revolves around 80%, while for the Weather Solar data this happens at F-Measure=90%. Thus, the accuracy of SWO-UnO only slightly differs upon altering the reduction ratio. As happens with low number of dimensions in SWO-MuO, SWO-UnO also exhibits fluctuations upon altering the window size between $\omega = 8$ to $\omega = 24$ and the $minSup$ from 5% to 10% upon using lower batch sizes. This is depicted in Intel data related Figures 7(b) and 7(c) where a low $\lambda = 4$ value is used. On the other hand, in Weather Solar, Figures 8(b) and 8(c), where $\lambda = 8$ the effect of the window size and the minimum support on F-Measure is not equivalently noteworthy. It is important to note that according to Lemma 3 and Lemma 4 the ratio $\frac{\omega}{\lambda}$ that affects the upper (probability) bound is the same in Intel and Weather SWO-UnO configurations. However, in practice it appears that low values of λ can have a stronger (negative) effect on the accuracy of SWO-UnO, since compressing smaller batches can lead to more evident discontinuities in similarity computations using the reduced bitmaps that operate in discrete spaces.

As in TWO-MuO and SWO-MuO, Figures 7(d) and 8(d) show that SWO-UnO exceeds its theoretically expected benefits as well. More precisely, in Intel Voltage SWO-UnO $1/8\,Reduction$ saves in total communication up to a factor of 3.37 instead of the expected 2.67, while in Weather Solar the total communication reduction reaches a factor of 5.43 instead of expectedly 4. At the same time and for the same reduction ratio, according to Figure 7(e), in Intel Voltage the network lifetime is prolonged by a factor of 3.8 instead of 2.67 (or the 3.37 reduction in transmitted bits). Furthermore, Figure 8(e) reports a 5.65 increment in network lifetime instead of expectedly 4 or 5.43 times i.e., the reduction in transmitted bits observed in Figure 8(d). An important observation is that in Weather Solar, due to the higher percentage of final outliers (categorized as $ToBS$), the $Inter-$cluster communication cost is higher than the $Intra-$cluster phase. Therefore, in this particular occasion, it appears preferable to choose $SelectStar$, instead of $NoReduction$, as the less naive approach which centralizes all the data immediately after the intracluster processing phase.



(a) SWO-UnO Weather: Reduction Ratio (b) SWO-UnO Weather: Window Size (c) SWO-UnO Weather: minSup

(d) SWO-UnO Weather: Communication Cost (e) SWO-UnO Weather: Energy Consumption

Figure 8: SWO-UnO Analysis on Weather Data

7. Applicability to Broader IoT Settings and Security Aspects

So far we focused purely on data management and (accuracy, communication) performance issues throughout the outlier detection process using our TWO-MuO, SWO-MuO, SWO-UnO operational modes. This section aims at discussing the applicability of our techniques in broad categories of IoT scenarios and their orthogonality with security and privacy challenges arising in IoT settings [39, 40, 41]. In our discussion we use the recent work of [41] as a guideline. More precisely, the discussion in [41] identifies key IoT application fields and shows how the clustering algorithm for anomaly detection proposed in [42] can be adapted to a number of different scenarios and act when security, privacy issues become of the essence. Major application domains of (cloud-integrated) IoT include, but are not limited to, Industry 4.0 settings [42, 43, 44], Power Grids [45] and Smart City [46] applications, Healthcare [47, 48, 49] and Environmental Monitoring [16]. Our experimental evaluation in the previous section already accounted for environmental and meteorological data and the detection of outliers on them using all our proposed operational modes. Therefore, in what follows we concentrate on the rest of the domains outlined above. We choose to discuss security and privacy aspects separately for ease of exposition and to better exhibit the ability of our techniques to adapt to such application needs.

Industry 4.0 settings: The term Industry 4.0 or Industrial Internet involves integration of complex physical machinery and devices with networked sensors. These sensors collect and convey data exploited to gain insights so as to better control business and production outcomes. Industrial Internet has a broader focus on many industries and application areas while Industry 4.0 mainly focuses on manufacturing and logistics. ”Smart Industry”, or ”Smart Manufacturing” are also used to describe similar application contexts.

The framework proposed in [42] tackles anomaly detection over industrial wireless sensor networks. The architecture of the assumed distributed industrial infrastructure, at its lower tier, is equivalent to our clustered sensor network setting. In particular it is composed of a sensor network consisting of plain motes, cluster headers, wireless links and a gateway node. To complete the picture, the gateway node is supposed to be connected to the industrial process control system, a HMI (Human Machine Interface) Server presents the updated information collected over the WSN, while a Historical Database [50] keeps a record of all aggregated observations.

The core of the technique proposed in [42] is a distributed, fuzzy data clustering algorithm aiming at accurately modeling the normal and abnormal behavior of the sensed data space. First, local (at the mote level) data partitioning is performed using fuzzy c-means (fcm) clustering, and statistical thresholds are adaptively introduced to classify the data points into clusters and identify local outliers using local correlations at each mote. Later, the resulting cluster

centroids and outliers are transmitted higher to the WSN hierarchy. In that, communication savings are obtained by having cluster centroids and ranges represent normal data points and only transmitting outlying measurements. At the next level of the hierarchy, data points are again evaluated for cluster memberships considering more global correlations on the union set of data points at each level. This is performed recursively until the gateway node enables global anomalies to be detected using the thresholds on membership values computed at this (global) level.

Our claims in Section 5 about the key points that differentiate the current work compared to related works still hold for [42]. Hence, here we go beyond guarantees and trade-offs on bandwidth vs accuracy and the support for various window models, i.e., the basic advantages of our algorithmic arsenal, in order to study the ability of our techniques to adapt to the rest of the issues covered in [42] and in [43, 44] which employ a similar to [42] rationale. In particular [43] discusses a distributed data clustering approach that directly considers information characteristics using an appropriate entropy measure and avoids predefining the number of expected clusters. [44] on the other hand, utilizes [42] towards identifying and by-passing areas of the network prone to producing peculiar measurements.

The technique of [42] identifies both local anomalies using similarities among data within a single node, whereas global anomalies are identified considering correlations on the higher levels of the network hierarchy. Our techniques can be adapted to detect outliers at different levels of granularity and globally. This can be achieved by forwarding towards the control process system and HMI server information about both the intra- (clusterhead) and inter-cluster (towards global) communication phase.

In Section 1 we argued about the necessity of judging outliers not by considering motes' local readings, but correlations of measurements of various sensors in space and time. However, if both global and local anomaly detection is desired, our approach can adapt to this case as well. TWO-UnO, TWO-MuO, SWO-MuO, SWO-UnO do not support the soft data partitioning of [42] or entropy-based cohesive region extraction in [43]. Excluding this, they can contribute in identifying local outliers by performing similarity checks taking into consideration windows of observations obtained locally by a mote and judging their similarities. The RHP-based compression of local measurements in these windows also aids in this scenario for saving memory resources. Local anomalies can be judged by our approaches employing a range of supported uni- or multi-dimensional similarity measures (Table 2 and Table 3) and under different window types. Remarkably, sliding windows over RHP bitmaps would not be possible to support before the contributions of SWO-MuO and SWO-UnO made in the current work.

The work in [44] describes a routing scheme that by-passes motes producing outliers, using a twin rolling balls technique. Using the proposed technique, [44] diverts the packets routed inside the identified area of the network covered by these motes. The identification of infected nodes is done by adapting the data clustering approach of [42] to classify motes based on the fraction of anomalous data that is detected in individual data streams. This information is then used in the proposed by-passed routing (BPR) scheme. The techniques proposed in our work do not adopt the same definition of infected mote and infected area as in [44]. Despite this difference, the outlier information provided by our techniques that reaches the HMI server and the process control system involves the motes producing such peculiar measurements. Therefore, accompanying outlier reports by outlier mote locations helps in isolating these infected motes. One step further, by examining spatial proximity of infected motes, infected areas covered by them can be identified as well. However, because our emphasis in this work is on data management and query answering rather than networking/routing issues, the information about extracted outliers is to provide alarms accompanied by the areas covered by outlying motes (e.g. an imminent breakdown of a production machine or problematic production line) or exclude their measurements from (aggregate or ad-hoc) query answers.

Healthcare Monitoring: Remote patient health monitoring using smart wearable sensors opens up new opportunities for vigilance, timely diagnosis and reaction in case of adverse events. For instance, let us consider a scenario similar to the one introduced in [47] where the monitoring procedure involves a stay-at-home patient to be monitored by body sensors. The measurements of the sensors are delivered to the patient's mobile phone device via Bluetooth and are subsequently communicated to hospital servers via Internet. Following [47], hospital servers then apply a clustering algorithm (a centralized version of [42]), define respective (normal and abnormal) clusters and their ranges, which are communicated back to the patient's mobile device. Based on this information, patient's mobile device keeps receiving updates from body sensors, compares these measurements with the identified clusters to detect and classify potential anomalies (please refer to [47, 48] for further details).

Body sensor measurements in such a scenario may involve Electrocardiography (ECG) signals which need to be compressed since they are enormously large in size. Thus, compression aims at both avoiding depleting the available

memory of the mobile device and to reduce communication costs during transmissions to the central server [47, 48]. The patient record analysis is not necessarily restricted to ECG, but as [49] argues may involve other monitored vital signs such as Systolic/Diastolic Blood Pressure (SBP/DBP), Respiratory Rate (RR), Blood O_2 saturation, among which discovering correlations is important for clinical decision-making. Decompressing these signals in any stage of this process is too time consuming and may affect the real-timeliness of the application. The latter is obviously of utmost importance. Therefore, if one has to detect anomalies on such signals, working on compressed representations is the only way to go.

Let us now examine how the above scenario, despite being deprived from in-network processing and pair-wise similarity checks among sensors, can be supported by our approaches. First, our algorithms are capable of detecting anomalies in both uni- and multi-dimensional setups. Therefore, they are applicable in case a single or multiple vital signs should be included in the analysis. Second, the support of different window models allows the application to tune the frequency by which both new clusterings will be produced as well as groups of signals will be classified as normal or abnormal. For instance, it may be desired to produce new clusterings after accumulating a number of recent measurements (tumble) per patient or continuously check for concept drifts in the produced clusterings in a sliding window fashion.

In all such cases, LSH acts as the compression enabler. Interestingly, the produced RHP representations do not need to be decompressed since our proposed techniques can be used to solve what can be thought of as the dual to the outlier detection problem, that of clustering. In such cases, compressed representations along with the desired supported measure (Table 2 and Table 3) can be used to compute approximate clusterings on the produced bitmaps. In fact, clustering based on LSH representations has been repeatedly applied in the literature [51]. Nonetheless, special care should be taken with respect to the accuracy of the produced clusterings and during the classification process. This is because, although the accuracy guarantees and the approximation quality of our techniques are tunable based on the chosen communication and memory utilization reduction ratio, in this particular application field False Negatives (FN) (i.e., classifying an anomaly as normal signal) may have severe consequences for the patient. On the contrary, a False Positive (FP) (i.e., classifying a normal signal as potentially abnormal) is much more tolerable from an application viewpoint. In order to better control the amount of FNs, an amplification process [52] can be applied at the sensor level (before the intra- and inter-cluster phases). This involves the construction of multiple bitmaps and the proper combination of logical conjunctions (*AND*) and disjunctions (*OR*) among them. In that, the bitmap that is transmitted by each mote to its clusterhead or among clusterheads results after the application of a series of *AND* and *OR* operations on multiple RHP applications (on the same vector) and their corresponding bitmaps. Hence, by choosing the right number and sequence of such operations one can successfully tune the resulted bitmaps and thus our operational modes towards a more FN-free version. Please refer to Chapter 3 of [52] for further details on amplification.

Power Grids: Smart grids are energy networks that can automatically monitor energy flows and adjust to changes in energy supply and demand accordingly. When coupled with smart metering systems, smart grids reach consumers and suppliers by providing information on real-time consumption. With smart meters, consumers can adapt - in time and volume - their energy usage to different energy prices throughout the day, saving money on their energy bills by consuming more energy in lower price periods [53].

The key to a continuous improvement of power supply and its use is up-to-date information about local system circumstances. This involves power monitoring, power quality recording, disturbance recorders and phasor measurement. Power Quality and Measurement supports providers and energy users with solutions for precise measurement, acquiring and reporting of necessary information so that the power system health can be determined, adapted and improved in a continuous process.

Wide Area Monitoring Systems (WAMS) are composed of Phasor Measurement Units (PMUs) installed in the substations of the power network, a Phasor Data Concentrator (PDC) server in the regional control center and a communication network to establish an online connection between substations and control center [54]. Modern compact and less expensive PMUs communicate the measured data back to the control room in real-time, eliminating the need for local storage and advanced user interface. All that is needed is a communication interface such as a wireless modem. This structure already resembles the two-tiered architecture we employed in our analysis, although PDC (clusterhead) determination and PMU (sensor) assignment to PDCs is performed under different criteria.

The data recorded is multi-dimensional, as each PMU records multiple time-series for each measurement type such as frequency, speed of frequency change and voltage. The availability of a very accurate clock through the Global Positioning System has made it possible to synchronize phasor measurement units over a large area which perfectly supports the data model described in Section 2.1 where motes obtain measurements in a synchronized way. What is more, a single PMU cannot detect anomalies in the grid. However, comparing simultaneous measurements of phasors at strategically-selected positions in the grid can reveal many problems and anomalies. This strengthens our arguments (Section 1) that outliers, and therefore potential anomalies in the grid, cannot be judged based on motes (PMUs in this scenario) local readings and correlations in space and time need to be accounted for. Eventually, the utility of the similarity measures employed by our techniques such as correlation coefficient in Table 2 and their multidimensional versions in Table 3 do make sense in PMU data analysis [45]. Evidently, RHP application in this scenario aims at reducing memory requirements and communication costs tuning the accuracy of the anomaly detection process at the same time.

Therefore, our above discussion absolutely demonstrates the applicability of our techniques for Smart Grid monitoring including the whole suite of proposed TWO-UnO, TWO-MuO, SWO-MuO, SWO-UnO operational modes.

Smart City Applications: A smart city is a concept which integrates multiple information and communication technology solutions to efficiently and effectively manage a city's assets. City assets include, but are not limited to, transportation systems, waste management, water management, safety systems, local departments information systems and other community services. Among these we are going to concentrate on anomaly detection in transportation systems since our previous discussion about other application domains is more easily adapted to the rest of the city assets cited above. A smart transportation system aims at providing intelligent services of transport and traffic management. In the latter context, semantic trajectory extraction becomes of the essence [55]. The focus of semantic trajectory construction is on the extraction of meaningful trajectories from the raw positioning data like GPS feeds and other attributes stemming, for instance, from sensory elements placed on vehicles. Such attributes may include vehicle heading, steering wheel activity, position in lane, distance to head-away vehicle etc. Semantic trajectories manage to encompass both objects' spatiotemporal movement characteristics as well as useful information regarding objects' movement patterns (e.g dwelling, speeding, tailgating). In other words, semantic trajectories constitute spatiotemporal trajectories broken down to semantically enriched parts with each part being tagged by a semantic label.

Outlier detection over semantic trajectories can be useful in many traffic monitoring scenarios where authorities are interested in identifying apart from recent objects' trajectory representation, the behavior of the drivers. In that, authorities are able to continuously diagnose drivers tending to have aggressive (speeding, tailgating, driving at the edges of the lanes etc.) behavior, thus creating alerts for patrol cars and enabling suitable placement and periodic rearrangement of traffic wardens.

The work of SeTraStream [55] was the first in the literature that described a framework for online semantic trajectory construction over streaming movement data. In a nutshell, SeTraStream centralizes vehicle information involving both GPS and non-GPS attributes from sensory devices attached on vehicles. It then attempts to extract in a per vehicle fashion, portions of the trajectory where the driving behavior is homogeneous or, differently put, points in time where the driving behavior changes significantly (called episodes). Thus, the latest part of a trajectory can be distinguished as semantically different from previous parts of the vehicle's movement, thus obtaining its own tag/label. The framework by itself does not work over a distributed setting. However, it argues about the suitability of a measure that is common in our analysis, namely the RV-Coefficient (Table 3), to be used for semantic trajectory construction upon judging points of semantic change.

Adapting our framework in an outlying semantic trajectory detection context is fairly intuitive. First, in our context we are not interested in working on a per vehicle basis, since detection of anomalies in an unsupervised manner (i.e. without having to define threshold values per road segment and time of the day) requires comparing the driving behavior of a vehicle to nearby ones to judge peculiar trajectory portions. Second, instead of being concerned with detecting episodes, we want to distinguish which trajectories are semantic outliers based on recent observations. Hence, sensor elements attached on vehicles periodically, using a tumbling window model, compress and transmit (GPS and non-GPS) attribute values to the nearest, stationary data aggregation point during the intracluster communication phase. Afterwards, aggregation points within a spatial vicinity in the city communicate during the intercluster communication phase to pinpoint outlying semantic trajectories based on RV-Coefficient similarity checks. In other

words, the only difference with respect to our TWO-MuO operational mode is that clusterheads are substituted by fixed, stationary aggregation points and at each intra- or inter-cluster communication phase, the aggregation point for each vehicle may change while it commutes. Nonetheless, for the same reason, our SWO-MuO approach is more difficult to apply since, due to mobility, different portions of the vectors of a sliding window for the same vehicle will be aggregated in different points. We reckon however, that for this specific scenario working with TWO-MuO's window model is very well acceptable from an application viewpoint.

Security and Privacy Considerations: Our reference point during the formulation of our above discussion, the work of [41], examines the ability of the clustering algorithm of [42] to adapt to scenarios where security aspects become of the essence. In particular, [41] discusses the security challenges that arise throughout the whole pipeline from data acquisition at the sensor level, data communication within the WSN and data processing at the end of the pipeline where a cloud infrastructure exists to perform computationally demanding, batch analytics. With respect to this pipeline, we here concentrate on the part involving privacy and security challenges at the WSN level, i.e., before having data available in the cloud platforms. This involves security at the sensor level and for data in transit (intra-, inter-cluster communication). The stage of the pipeline involving cloud integration is discussed in [41] and also covered in [40, 39]. Moreover, the work of [46] touches upon privacy issues in Smart City applications described earlier in this section. Secure and Privacy-preserving data mining approaches can be roughly classified in main categories including: randomization and cryptographic techniques. We here outline why and how our algorithmic suite can remain applicable even when security and privacy application requirements exist. Our goal, of course, is not to develop full fledged security and privacy protocols, but instead show how our algorithmic suite provides the primitives of operating orthogonally to such protocols.

Randomization is a data perturbation technique where the data distortion is masked by random data. The basic component for computing similarities among mote measurements in all operational modes (TWO-UnO, TWO-MuO, SWO-MuO, SWO-UnO) of our algorithmic suite is the computation of pairwise angle or, equivalently, cosine similarities using RHP application to reduce communication and tune accuracy. The work of [56] analyzes cosine similarity and combines it with obfuscation mechanisms in order to achieve a privacy preserving similarity computation solution. According to [56] there are two randomization means that should be used to enhance cosine similarity computation with privacy aspects. First, it shows that the multiplication of a (mote in our study) vector with a rotation matrix preserves cosine similarity. In other words, for a properly dimensioned rotation matrix M, $cos(a, b) = cos(M \cdot a, M \cdot b)$. Moreover, the random scaling of two vectors with different random scaling factors m_1 and m_2 preserves its cosine similarity, i.e. $cos(a, b) = cos(m_1 \cdot a, m_2 \cdot b)$. Such randomization aspects are orthogonal to all the operational modes included in our algorithmic suite. The idea is to apply these transformations to original mote data vectors or data matrices so that simultaneously angle/cosine of mote pairs is preserved and privacy is assured. In particular, at the sensor level, while sensors obtain measurements from the sensed environment, they can instantly apply vector scaling using arbitrarily chosen, different random vectors (m_i for mote S_i). Having done that on a per measurement basis, then, forming a tumble in TWO-UnO and TWO-MuO, a column of attribute values in SWO-MuO or a batch in SWO-UnO, motes can apply a vector rotation operation utilizing a common matrix M. Subsequently, RHP application involves an additional randomization step in which LSH bitmaps are produced before being transmitted in all the proposed operational modes. Besides the protocol of [56], the works in [57, 58] present more recent advancements in privacy-aware cosine similarity computation.

For TWO-UnO and TWO-MuO operational modes, the privacy aware RHP bitmaps produced using the procedure described above is the only data (besides mote ids, support, location and other potential meta-data) that needs to be communicated from motes to clusterheads (intracluster phase) and among clusterheads (intercluster phase). Nevertheless, recall that computing similarities in SWO-MuO and SWO-UnO entails the transmission of additional data regarding vector norms. Therefore, besides the above steps towards private similarity estimation, we complement our discussion with encryption aspects. As shown in Section 4.2 and Equations 9-12 in particular, the operations applied at the intra- and inter-cluster level to extract pairwise similarities involve simple additions and multiplications. This is also true for SWO-UnO in Section 4.3 which inherits SWO-MuO properties in Equations 13-15. Homomorphic encryption (HE) possesses the ability to serve high security standards by encrypting the data (mote vector norms, support values etc in our scenarios), but preserve the ability to perform operations (addition, subtraction, multiplication) directly on encrypted data as if it is performed on original data. In our resource constrained environment we are especially interested in lightweight HE schemes that will not deplete motes' residual energy. Moreover, because of the

aforementioned equations in our SWO-MuO, SWO-UnO modes we should choose lightweight schemes that support both addition and multiplication on encrypted data. According to [41, 40] Domingo-Ferrer's scheme satisfies this pair of properties, while [40] builds a cloud-based data processing model using it [6]. Hence, in SWO-MuO and SWO-UnO, at the sensor level norm values can be computed on the fly, before column vectors or batches undergo randomization, get encrypted and be transmitted to the clusterhead. Then, clusterheads use the encrypted norm values, encrypt the (extracted using the privacy enhanced RHP bitmaps according to our discussion in the previous paragraph) similarity values for each pair of tested motes and apply the result to Equations 9-12 or Equations 13-15 for SWO-MuO and SWO-UnO, respectively. Such lightweight HE schemes can also be used in all operational modes so as to encrypt query parameters, minimum support values per mote or other meta-data.

In conclusion, as our above extensive discussion demonstrates, the algorithmic suite we propose in this work finds applications in all major categories of IoT scenarios leveraging spatiotemporally referenced mote measurements. Furthermore, our techniques can be exploited under privacy and security constraints, at least to the extend we exposed above, to prevent adversaries accessing sensitive mote information both at the level of the mote and for data in-transit. Thus, the discussion in this section indeed strengthens our claim of providing an omnibus outlier detection solution in WSN settings.

8. Conclusions

The TACO framework (TWO-UnO operational mode in our work) initially introduced in [4, 5] was the first in the literature of outlier detection techniques in wireless sensor networks to provide a direct way of trading bandwidth for accuracy during the outlier detection procedure in a straightforward manner. In the current work we successfully confronted two important limitations of the initial framework regarding sliding window as well as multidimensional outlier detection support. We ended up with a novel framework capable of supporting four operational modes during outlier detection, which ensure its applicability to a wide spectrum of application needs. Thus, we indeed introduced a suite of methods composing an omnibus outlier detection solution. Apart from doing so, we reasoned about the ability of our framework to support a variety of similarity measures and we extensively elaborated on the outlier quality, for all the presented operational modes, versus the consumed bandwidth resources. We presented an extensive experimental analysis which proved the ability of the introduced TWO-MuO, SWO-MuO and SWO-UnO operational modes to significantly reduce communication and energy consumption while achieving high accuracy during the outlier detection process. Eventually, we elaborated on the applicability of our techniques in broader IoT (Industry 4.0, healthcare, smart grid, smart city, environmental monitoring) scenarios and their ability to satisfy privacy/security requirements.

Generic Conclusions: the accuracy of our techniques mostly remains resilient to high compression ratios (1/8, 1/16 etc) maximizing the potential for bandwidth preservation and network lifetime maximization. These savings come both during the intra- and inter-cluster communication phase of our algorithms in all operational modes. The latter exhibits the effectiveness of the in-network outlier detection procedure we foster. Remarkably, in realistic scenarios, under message collisions and retransmissions, the bandwidth and energy savings provided by our techniques exceed their theoretically expected values, irrespectively of the utilized operational mode. This is both because compressed data consume less bandwidth during retransmissions and also because they occupy communication channels for shorter time slots, thus reducing collisions during transmissions by multiple motes.

Operational Mode-Specific Conclusions: in theory, our proposed operational modes are expectedly more accurate with more strict outlier definitions expressed by tighter similarity thresholds. In practice, this behavior is more evident in SWO-MuO and SWO-UnO due to the fact that potentially "bad" RHP vectors are reused in window slides. On the contrary, TWO-MuO exhibits steady accuracy even with looser outlier definitions. The accuracy of SWO-MuO or SWO-UnO in theory is to deteriorate with larger window sizes. However, in practice this rarely happens, due to the fact that the more observations we include in the window, the more distinguishable sensors producing outlying measurements become. SWO-MuO, SWO-UnO provide lower, compared to TWO-MuO, but still significant communication savings. These savings are for SWO-MuO more dependent on the cardinality of the monitored attribute set,

[6] Lightweight Attribute-Based Encryption (ABE) scheme proposals tailored for IoT exist in the literature [59], but their focus is role-based access control on the data. Therefore, they do not fit our focus in this section.

while for SWO-UnO this dependence involves the fostered batch size. In other words, SWO-MuO's and SWO-UnO's communication and energy savings are maximized the higher the attribute set cardinality or the batch size becomes, respectively.

Rules of Thumb: if the application field requires multidimensional outlier detection and can tolerate a more "lazy" (tumble adoption) outlier detection scheme, the TWO-MuO operational mode with a maximum (1/8, 1/16 or greater) compression ratio is the right way to go. TWO-MuO is the most easily parameterizable operational mode since its accuracy is not significantly affected by the rest of the chosen parameters (similarity threshold, minimum support, tumble size). When multidimensional outliers need to be delivered in a continuous fashion with high frequency, SWO-MuO is the suitable alternative. In looser outlier definitions posed by the application field, to improve the accuracy of SWO-MuO, it should preferably be used along with a more fine-tuned selection of RHP vectors [27] and a greater (e.g. 8% or 10%) minimum support, irrespectively of the chosen sliding window size. Similar rules can be extracted for the unidimensional case, i.e., among TWO-UnO introduced in prior work [4, 5] and SWO-UnO discussed here. Nevertheless, in general, SWO-UnO is accurate (compared to SWO-MuO due to higher attribute homogeneity in the unidimensional case) even for looser similarity thresholds. It thus avoids the need for fine-tuned selection of RHP vectors. Both in SWO-MuO, SWO-UnO 1/8, 1/16 or greater compression ratio should be used to maximize network lifetime with only slight effect on accuracy.

Acknowledgment

We acknowledge support of this work by the project Moving from Big Data Management to Data Science (MIS 5002437/3) which is implemented under the Action Reinforcement of the Research and Innovation Infrastructure, funded by the Operational Programme "Competitiveness, Entrepreneurship and Innovation" (NSRF 2014-2020) and co-financed by Greece and the European Union (European Regional Development Fund). Yannis Kotidis's work was financed by the Research Center of Athens University of Economics and Business, in the framework of the project entitled 'Original Scientific Publications'.

References

[1] S. Madden, M. J. Franklin, J. M. Hellerstein, W. Hong, Tag: A tiny aggregation service for ad-hoc sensor networks, ACM SIGOPS Operating Systems Review 36 (SI) (2002) 131–146.

[2] A. Deligiannakis, Y. Kotidis, V. Vassalos, V. Stoumpos, A. Delis, Another outlier bites the dust: Computing meaningful aggregates in sensor networks, in: Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on, IEEE, 2009, pp. 988–999.

[3] S. Burdakis, A. Deligiannakis, Detecting outliers in sensor networks using the geometric approach, in: Data Engineering (ICDE), 2012 IEEE 28th International Conference on, IEEE, 2012, pp. 1108–1119.

[4] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, Y. Theodoridis, Taco: tunable approximate computation of outliers in wireless sensor networks, in: Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, ACM, 2010, pp. 279–290.

[5] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, Y. Theodoridis, In-network approximate computation of outliers with quality guarantees, Information Systems 38 (8) (2013) 1285–1308.

[6] N. Giatrakos, Y. Kotidis, A. Deligiannakis, Pao: power-efficient attribution of outliers in wireless sensor networks, in: Proceedings of the Seventh International Workshop on Data Management for Sensor Networks, ACM, 2010, pp. 33–38.

[7] Y. Zhang, N. Meratnia, P. Havinga, Outlier detection techniques for wireless sensor networks: A survey, IEEE Communications Surveys & Tutorials 12 (2) (2010) 159–170.

[8] S. Subramaniam, T. Palpanas, D. Papadopoulos, V. Kalogeraki, D. Gunopulos, Online outlier detection in sensor data using non-parametric models, in: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 187–198.

[9] U. Srivastava, K. Munagala, J. Widom, Operator placement for in-network stream query processing, in: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, ACM, 2005, pp. 250–258.

[10] M. S. Charikar, Similarity estimation techniques from rounding algorithms, in: Proceedings of the thiry-fourth annual ACM symposium on Theory of computing, ACM, 2002, pp. 380–388.

[11] M. Chatterjee, S. K. Das, D. Turgut, Wca: A weighted clustering algorithm for mobile ad hoc networks, Cluster computing 5 (2) (2002) 193–204.

[12] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: System sciences, 2000. Proceedings of the 33rd annual Hawaii international conference on, IEEE, 2000, pp. 10–pp.

[13] M. Qin, R. Zimmermann, Vca: An energy-efficient voting-based clustering algorithm for sensor networks., J. UCS 13 (1) (2007) 87–109.

[14] O. Younis, S. Fahmy, Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach, in: INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, Vol. 1, IEEE, 2004, p. 640.

[15] D. Carney, U. Çetintemel, M. Cherniack, C. Convey, S. Lee, G. Seidman, N. Tatbul, S. Zdonik, M. Stonebraker, Monitoring streamsa new class of data management applications, in: VLDB'02: Proceedings of the 28th International Conference on Very Large Databases, Elsevier, 2002, pp. 215–226.

[16] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, D. Culler, An analysis of a large scale habitat monitoring application, in: Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, SenSys '04, ACM, New York, NY, USA, 2004, pp. 214–226.
URL http://doi.acm.org/10.1145/1031495.1031521
[17] J. Ramsay, J. ten Berge, G. Styan, Matrix correlation, Psychometrika 49 (1984) 403–423.
[18] J. Josse, J. Pagès, F. Husson, Testing the significance of the rv coefficient, Comput. Stat. Data Anal. 53 (2008) 82–91.
[19] M. X. Goemans, D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, Journal of the ACM (JACM) 42 (6) (1995) 1115–1145.
[20] G. Xue, Y. Jiang, Y. You, M. Li, A topology-aware hierarchical structured overlay network based on locality sensitive hashing scheme, in: Proceedings of the second workshop on Use of P2P, GRID and agents for the development of content networks, ACM, 2007, pp. 3–8.
[21] B. Karp, H.-T. Kung, Gpsr: Greedy perimeter stateless routing for wireless networks, in: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, 2000, pp. 243–254.
[22] M. Garofalakis, J. Gehrke, R. Rastogi, Querying and mining data streams: you only get one look a tutorial, in: Proceedings of the 2002 ACM SIGMOD international conference on Management of data, ACM, 2002, pp. 635–635.
[23] C. Giannella, J. Han, J. Pei, X. Yan, P. S. Yu, Mining Frequent Patterns in Data Streams at Multiple Time Granularities, MIT Press, 2002.
[24] T. Palpanas, Real-time data analytics in sensor networks, in: Managing and Mining Sensor Data, Springer, 2013, pp. 173–210.
[25] M. Gabel, A. Schuster, D. Keren, Communication-efficient distributed variance monitoring and outlier detection for multivariate time series, in: Parallel and Distributed Processing Symposium, 2014 IEEE 28th International, IEEE, 2014, pp. 37–47.
[26] J. Wang, H. T. Shen, J. Song, J. Ji, Hashing for similarity search: A survey, CoRR abs/1408.2927.
URL http://arxiv.org/abs/1408.2927
[27] K. Georgoulas, Y. Kotidis, Distributed Similarity Estimation using Derived Dimensions, The VLDB Journal 21 (1) (2012) 25–50.
URL http://dx.doi.org/10.1007/s00778-011-0233-y
[28] S. R. Jeffery, G. Alonso, M. J. Franklin, W. Hong, J. Widom, Declarative support for sensor data cleaning, in: International Conference on Pervasive Computing, Springer, 2006, pp. 83–100.
[29] E. Elnahrawy, B. Nath, Cleaning and querying noisy sensors, in: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, ACM, 2003, pp. 78–87.
[30] F. Chu, Y. Wang, D. S. Parker, C. Zaniolo, Data cleaning using belief propagation, in: Proceedings of the 2nd international workshop on Information quality in information systems, ACM, 2005, pp. 99–104.
[31] S. R. Jeffery, M. Garofalakis, M. J. Franklin, Adaptive cleaning for rfid data streams, in: Proceedings of the 32nd international conference on Very large data bases, VLDB Endowment, 2006, pp. 163–174.
[32] J. Chen, S. Kher, A. Somani, Distributed fault detection of wireless sensor networks, in: Proceedings of the 2006 workshop on Dependability issues in wireless ad hoc networks and sensor networks, ACM, 2006, pp. 65–72.
[33] X.-Y. Xiao, W.-C. Peng, C.-C. Hung, W.-C. Lee, Using sensorranks for in-network detection of faulty readings in wireless sensor networks, in: Proceedings of the 6th ACM international workshop on Data engineering for wireless and mobile access, ACM, 2007, pp. 1–8.
[34] Y.-J. Wen, A. M. Agogino, K. Goebel, Fuzzy validation and fusion for wireless sensor networks, in: ASME 2004 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, 2004, pp. 727–732.
[35] A. Deligiannakis, Y. Kotidis, N. Roussopoulos, Dissemination of compressed historical information in sensor networks, The VLDB Journal 16 (4) (2007) 439–461.
URL http://dx.doi.org/10.1007/s00778-005-0173-5
[36] 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.
[37] H. O. Tan, I. Körpeoğlu, Power efficient data gathering and aggregation in wireless sensor networks, SIGMOD Rec. 32 (4) (2003) 66–71.
URL http://doi.acm.org/10.1145/959060.959072
[38] F. Fazel, M. Fazel, M. Stojanovic, Random access compressed sensing for energy-efficient underwater sensor networks, IEEE Journal on Selected Areas in Communications 29 (8) (2011) 1660–1670.
[39] A. Alabdulatif, I. Khalil, M. Reynolds, H. Kumarage, X. Yi, Privacy-preserving data clustering in cloud computing based on fully homomorphic encryption, in: PACIS 2017: Societal Transformation Through IS/IT, Association for Information Systems (AIS), 2017, pp. 1–13.
[40] A. Alabdulatif, H. Kumarage, I. Khalil, X. Yi, Privacy-preserving anomaly detection in cloud with lightweight homomorphic encryption, Journal of Computer and System Sciences 90 (2017) 28–45.
[41] H. Kumarage, I. Khalil, A. Alabdulatif, Z. Tari, X. Yi, Secure data analytics for cloud-integrated internet of things applications, IEEE Cloud Computing 3 (2) (2016) 46–56.
[42] H. Kumarage, I. Khalil, Z. Tari, A. Zomaya, Distributed anomaly detection for industrial wireless sensor networks based on fuzzy data modelling, Journal of Parallel and Distributed Computing 73 (6) (2013) 790–806.
[43] H. Kumarage, I. Khalil, Z. Tari, Granular evaluation of anomalies in wireless sensor networks using dynamic data partitioning with an entropy criteria, IEEE Transactions on Computers 64 (9) (2015) 2573–2585.
[44] N. Yaakob, I. Khalil, H. Kumarage, M. Atiquzzaman, Z. Tari, By-passing infected areas in wireless sensor networks using bpr, IEEE transactions on computers 64 (6) (2015) 1594–1606.
[45] H. Jiang, K. Wang, Y. Wang, M. Gao, Y. Zhang, Energy big data: A survey, IEEE Access 4 (2016) 3844–3861.
[46] A. Alabdulatif, I. Khalil, H. Kumarage, A. Y. Zomaya, X. Yi, Privacy-preserving anomaly detection in the cloud for quality assured decision-making in smart cities, Journal of Parallel and Distributed Computing.
[47] F. Sufi, I. Khalil, Diagnosis of cardiovascular abnormalities from compressed ecg: a data mining-based approach, IEEE transactions on information technology in biomedicine 15 (1) (2011) 33–39.
[48] F. Sufi, I. Khalil, A. N. Mahmood, A clustering based system for instant detection of cardiac abnormalities from compressed ecg, Expert Systems with Applications 38 (5) (2011) 4705–4713.
[49] A. R. M. Forkan, I. Khalil, A clinical decision-making mechanism for context-aware and patient-specific remote monitoring systems using the correlations of multiple vital signs, Computer methods and programs in biomedicine 139 (2017) 1–16.

[50] J. Brito, K. Demirkaya, B. Etienne, Y. Katsis, C. Lin, Y. Papakonstantinou, Efficient approximate query answering over sensor data with deterministic error guarantees, arXiv preprint arXiv:1707.01414.
[51] X. Jin, J. Han, Locality sensitive hashing based clustering, in: Encyclopedia of Machine Learning, Springer, 2011, pp. 613–613.
[52] J. Leskovec, A. Rajaraman, J. D. Ullman, Mining of massive datasets, Cambridge university press, 2014.
[53] Smart Grids Task Force, Smart grids and meters (accessed February 15, 2018).
URL https://ec.europa.eu/energy/en/topics/markets-and-consumers/smart-grids-and-meters
[54] Phasor Measurement Unit (PMU) and Grid Monitoring (accessed February 15, 2018).
URL https://w3.siemens.com/smartgrid/global/en/products-systems-solutions/Protection/pmu-phasor-measurment-unit/Pages/pmu-phasor-measurement-unit.aspx
[55] Z. Yan, N. Giatrakos, V. Katsikaros, N. Pelekis, Y. Theodoridis, Setrastream: semantic-aware trajectory construction over streaming movement data, in: International Symposium on Spatial and Temporal Databases, Springer, 2011, pp. 367–385.
[56] I. Leontiadis, M. Onen, R. Molva, M. J. Chorley, G. B. Colombo, Privacy preserving similarity detection for data analysis, in: Cloud and Green Computing (CGC), 2013 Third International Conference on, IEEE, 2013, pp. 547–552.
[57] R. Lu, H. Zhu, X. Liu, J. K. Liu, J. Shao, Toward efficient and privacy-preserving computing in big data era, IEEE Network 28 (4) (2014) 46–50.
[58] Z. Gheid, Y. Challal, An efficient and privacy-preserving similarity evaluation for big data analytics, in: Utility and Cloud Computing (UCC), 2015 IEEE/ACM 8th International Conference on, IEEE, 2015, pp. 281–289.
[59] X. Yao, Z. Chen, Y. Tian, A lightweight attribute-based encryption scheme for the internet of things, Future Generation Computer Systems 49 (2015) 104–112.