Information Sciences 467 (2018) 149–162

Contents lists available at ScienceDirect

Information Sciences

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

Characterizing unpredictable patterns in Wireless Sensor Network data

Luca Cagliero, Tania Cerquitelli*, Silvia Chiusano, Paolo Garza, Antonio Attanasio

Dipartimento di Automatica e Informatica, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino 10129, Italy

ARTICLE INFO

Article history:
Received 7 April 2015
Revised 10 May 2018
Accepted 1 August 2018
Available online 2 August 2018

Keywords:
Data mining
Infrequent weighted itemsets
Wireless sensor data

ABSTRACT

Wireless Sensor Network (WSN) monitoring takes a primary role in many industrial and research processes. Huge amounts of WSN sensor readings are nowadays available and can be analyzed to discover fruitful knowledge.

This paper focuses on analyzing historical WSN sensor readings to identify the combinations of sensors whose readings show an unexpected trend. Although significant variations of single sensor readings may be easily detected, discovering correlations between multiple sensor readings is challenging without using advanced data analytics tools. To tackle this issue, we present an itemset-based data mining approach to analyzing WSN data. It identifies the combinations of sensors (of arbitrary size) whose readings are unexpectedly low in a given time period. Since the readings acquired by multiple sensors may decrease in an alternate fashion, the discovered patterns provide new information compared to single sensor analysis. To make the mined patterns manageable by domain experts for manual inspection, the mining algorithm is driven by spatial constraints defined on the WSN topology.

The experimental results, achieved on real WSN data, demonstrate the effectiveness of the proposed approach in detecting heating system malfunctioning.

1. Introduction

In the last few years, the interest in Wireless Sensor Networks (WSN) has continuously grown in both the industrial and research fields. The attention has been focused on the design, implementation, and exploitation of novel technologies and applications to support WSN management (e.g., sensor interface [33], network architecture [27,41], integration of WSN and RFID technologies [38]). A parallel interest has been devoted to developing novel analytics tools to gain insights into WSN data [5,7,13,23,24,35]. Following these trends, in recent years many multi-utility companies have deployed WSNs based on sensors and smart meters to remotely control the provided services [1,2,9,25]. For example, companies operating in the sectors of electricity, thermal energy for district heating, gas, management of integrated water services, and waste collection and disposal all need to monitor WSNs.

Wireless Sensor Networks monitor the environment in different time slots and periods, even when inhabitants or visitors are not present (e.g., during nighttime or holidays). Detecting anomalous behaviors may help domain expert to prevent damages, wastes, or malfunction of electronic or electrical systems. Let us consider, for example, a server farm hosting tens

* Corresponding author.
E-mail addresses: luca.cagliero@polito.it (L. Cagliero), tania.cerquitelli@polito.it (T. Cerquitelli), silvia.chiusano@polito.it (S. Chiusano), paolo.garza@polito.it (P. Garza), antonio.attanasio@polito.it (A. Attanasio).

https://doi.org/10.1016/j.ins.2018.08.002
0020-0255/© 2018 Published by Elsevier Inc.

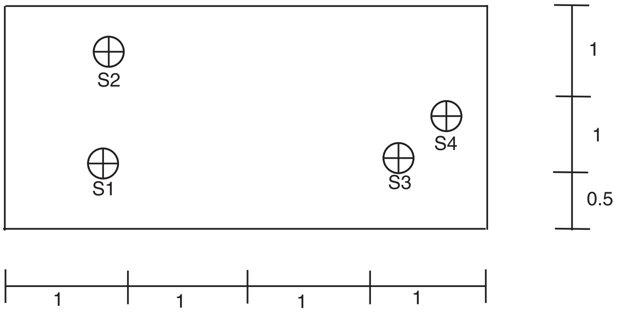


Fig. 1. Toy example of WSN topology.

or hundreds of servers in air-conditions environments. Capturing anomalous situations in the temperature measurements acquired by the WSN can prevent damages to the hardware.

Even though WSNs allow providers to effectively monitor different phenomena, little profits can be gained from WSN data unless discovering actionable knowledge from data. As a matter of fact, few companies analyze their owned data to support decision-making [21]. Since the largest part of WSN data remain unused, companies get a late and incomplete feedback on the evolution of the industrial processes and/or on the quality of the offered services. Hence, there is a need for novel and effective analytics systems aimed to support business decisions through the advanced analysis of historical WSN data. Since the number of sensor readings continuously grows, an increasing research interest has been devoted to applying data mining techniques to analyze WSN data (e.g., [26,32,40]). In this paper we analyze the historical readings of WSN sensors with the goal of identifying the combinations of sensors whose reading values show an unexpected trend. The position of this work with respect to the existing literature is discussed in Section 2.

Example. Let us consider a WSN deployed in a workplace to monitor the environmental temperature. The topology of the network is depicted in Fig. 1. It consists of four sensors s_j $(1 \le j \le 4)$. The acquired readings of the sensors in the network are collected into a relational dataset. For the sake of simplicity, let us consider the extract of the dataset reported in Table 1, which collects the temperature measurements acquired by each sensor at 6 sampling time instants t_i $(1 \le i \le 6)$. For example, at time t_1 sensor s_2 read 27 °C.

We look for the combinations of sensors whose temperature readings are relatively low in a given time period. These combinations may indicate a malfunction of the heating system. Notice that the problem we address in the context of heating system maintenance is common in other contexts, such as smart city monitoring. For example, if the topology would represent a city map and sensors would measure the environmental conditions in different city areas (e.g., temperature, pressure) a similar analysis can be performed to study the impact of pollutant agents on the environmental conditions. Experts may manually explore WSN data to analyze the temperature readings of individual sensors.

Example. Both sensors s_1 and s_2 measured an average temperature above 19 °C over all the sampling time instants (i.e., 19.9 °C for both sensors). Thus, apparently, they do not show any malfunction of the heating system. However, analyzing the correlation between the temperature measurements acquired by the two sensors it turns out that the reading of one of the two sensors in the set $\{s_1, s_2\}$ is, on average, less than or equal to 19 °C over all the sampling time instants. Specifically, at each sampling instant the least temperature reading acquired by sensors s_1 and s_2 is 15 °C at times t_1 (by sensor s_1), t_2 (s_1), t_3 (s_2), and t_4 (s_2); 22 °C at time t_5 (s_2); 20 °C at time t_6 (s_1). The average least temperature computed over all the sampling time instants is 17 °C, which is below the average temperature of livable places. The automatic discovery of potentially critical sets of sensors helps domain experts to drive monitoring activities. For instance, in light of the achieved results, the reasons behind the unexpected behavior of sensors s_1 and s_2 can be further investigated using domain-specific knowledge.

This paper proposes a novel type of pattern to automatically discover the sets of sensors in the WSN showing unexpected behaviors. This pattern, named *unexpected pattern*, represents an arbitrary set of sensors whose readings are unexpectedly low during the considered time period. To the best of our knowledge, this work is the first attempt to discover this type of pattern from WSN data. The proposed pattern allows us to discover trends in WSN data that cannot be discovered based on single sensor analysis.

Example. Unexpected pattern $\{s_1, s_2\}$ contains at least one sensor reading that is, on average, less than or equal to a given threshold (19 °C) over all the sampling time instants. A separate analysis of sensors s_1 and s_2's readings is not sufficient to identify a similar unexpected behavior.

Unexpected patterns can be specialized by enforcing constraints derived from the spatial topology of the WSN. Specifically, two main categories of patterns have been identified: (i) patterns satisfying the *closeness constraint*, which consist of a set of nearby sensors, and (ii) patterns satisfying the *distance constraint*, which consist of a set of distant sensors. On the one hand, to optimize the position of the sensors, experts may analyze the readings of nearby sensors. On the other hand, to highlight unexpected behaviors in WSNs they may analyze the correlation between distant sensor readings. To customize the analyses on different use cases, an extended version of a state-of-the-art itemset mining algorithm [10], which integrates the newly proposed constraints, has been proposed. According to the selected use case, only the combinations of nearby/distant sensors are extracted. Therefore, the mining result contains only the patterns that are worth considering for targeted analyses.

The proposed approach currently relies on in-memory data analyses. Since WSN data are expected to continuously grow in the number of historical sensor readings, a parallel issue is the extension of the proposed approach in a Big Data scenario. To address this issue, this paper also envisages perspectives of extensions of the proposed approach towards a distributed architecture.

The effectiveness of the proposed approach in detecting heating system malfunctioning was evaluated on real data acquired from a WSN located in a University campus network.

This paper is organized as follows. Section 2 compares this work with the existing literature. Sections 3 and 4 thoroughly describe the proposed approach and summarize the main experimental results, respectively. Section 5 discusses the perspectives of extension of the proposed system towards a scalable service, while Section 6 draws conclusions and discusses future works.

2. Related works

A relevant research effort has been devoted to applying existing data mining algorithms to WSN data. They addressed

(i) *WSN data clustering* [5,26,40], to group sensors acquiring correlated readings with the aim at reducing network communication costs,

(ii) *WSN data classification* [14,15,34], to predict the class of heterogeneous sensor readings [14,34] or to approximate the reading values of nearby sensors [15], and

(iii) *Pattern mining from WSN data* [8,11,31,42], to study the underlying correlations between sensor readings.

This paper addresses the problem of pattern mining from WSN data. Hence, it belongs to category (iii).

Recent pattern mining approaches tailored to WSN data entail

(1) Discovering *spatial correlations* between sensors [8], and

(2) Mining *temporal correlations* from data streams acquired from WSNs [31]. In [8,31] the authors extracted frequent patterns to characterize WSN data. Sensors were deemed as correlated if they acquired similar measurements in the analyzed time period. The discovered correlations between sensor readings highlight common behaviors, which can be useful for resizing or reconfiguring the network. In contrast, this work addresses the complementary goal of discovering unexpected behaviors in WSNs. The patterns proposed in this work represent the combinations of sensors whose readings are significantly different. Since we look for anomalous situations, the mined patterns rarely occur in the analyzed data. To extract them, we exploited an infrequent itemset mining algorithm.

An attempt to use infrequent patterns in WSN data analysis has been made in [42]. The authors focused on detecting salient events by discovering patterns which represent series of sensor readings. Per-sensor readings are processed in real time and then transmitted to a fusion center. Unlike [42], in this work we do not study the sequences of individual sensor readings, but the correlations among multiple sensors. To tackle this issue, sensor readings are processed offline.

A parallel research issue is the study of new itemset mining algorithms. To the best of our knowledge, the algorithms presented in [10,17,29] is the most recent solutions to the problem of discovering infrequent itemsets. In this work, we exploit an infrequent itemset mining algorithm to extract the unexpected patterns from WSN data. Among the existing solutions, we applied the MIWI algorithm [10], because it can be easily customized to WSN data. Furthermore, we customized the MIWI mining process to discover only the patterns of interest according to the WSN topology.

3. Wireless Sensor Network Analyzer

Wireless Sensor Network Analyzer (WSNA) is new data-driven approach to analyzing sensor readings acquired by Wireless Sensor Networks (WSNs). The analytical flow of WSNA is depicted in Fig. 2. A WSN acquires measurements through sensors distributed across the monitored environment. Sensor readings are prepared to the next analytical processes and then collected into a unique data repository. Next, an itemset mining technique is applied to the prepared dataset to discover unexpected behaviours in sensors' readings. The significance of the extracted patterns is manually validated by human experts based on their domain-specific knowledge. For example, a physical model of the environment where sensors are placed can be developed and exploited to assess the significance of the mined patterns.

A more detailed description of the steps of data preparation and pattern mining is given in the following sections.

3.1. Wireless sensor data preparation

Monitoring Wireless Sensor Networks (WSNs) entails collecting all the measurements of interest. A WSN can be modeled as a topology of sensors [30]. Let us denote as $S = \{s_1, \ldots, s_N\}$ the set of sensors in the topology. Each sensor is characterized by a unique identifier and by its geographic coordinates (i.e., latitude and longitude). For example, the WSN topology depicted in Fig. 1 consists of 4 sensors. For the sake of simplicity, let us suppose that the network topology is given by construction and it does not change over time, i.e., the sensor coordinates are fixed.

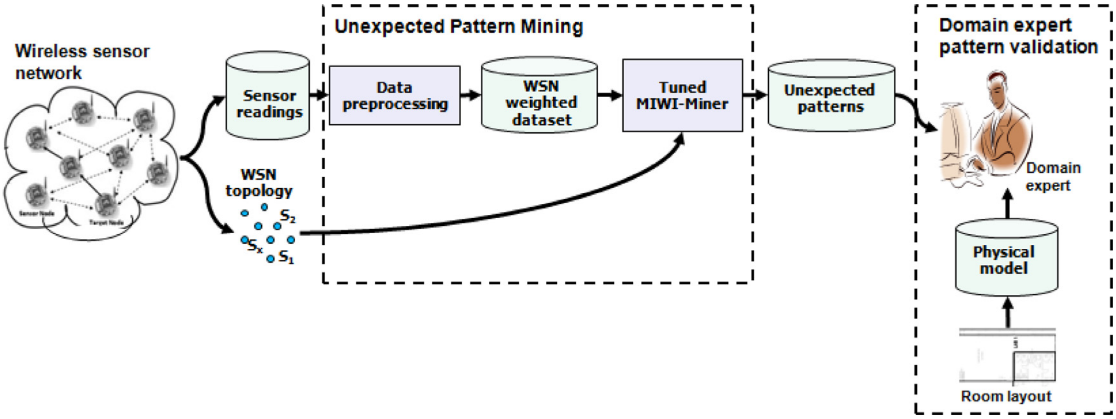


Fig. 2. Wireless Sensor Network Analyzer (WSNA).

To perform advanced data analyses, sensor readings are usually scheduled at a fixed rate and then stored into a centralized data repository [15,22]. Hence, a uniform temporal sampling of the physical measurement of interest is performed. At each sampling time, the readings acquired by all the sensors in the network are considered. Note that the phenomena monitored by WSNs often tend to have unknown spatial distributions, which potentially change over time. For this reason, changes in the topology (e.g., node suppression) or in the sampling schema are sometimes needed. The readings acquired with different sampling schemata should be separately analyzed to avoid introducing bias in the mining results. Possibly, a new mining session should be scheduled as soon as the sampling schema or the topology changes.

A WSN dataset collects all the sensor readings for a given measure m (e.g., temperature, humidity). To suit WSN data to the itemset mining process, the WSN dataset is tailored to a transactional data format. The concept of weighted item [39] is exploited to map sensors with the corresponding measurements. Specifically, for each sampling time instant t_i a weighted item associates a sensor s_j with its current measurement v_{ji}. In the context of WSNs, the concepts of item, weighted item, and WSN dataset can be formalized as follows.

Definition 1. *Item and weighted item.* Let M be a set of measures, S be the set of sensors in a WSN, and T be a set of points of time. Given a measure $m \in M$ (measured by sensors in S) and V a set of measurements for m, a function f_m: $S \times T \rightarrow V$ is defined. The following definitions hold. (i) Every sensor $s_j \in S$ is an *item*. (ii) a *weighted item* is a pair $\langle s_j, v_{ji} \rangle$, where the measurement v_{ji} is the weight associated with sensor s_j at time t_i, i.e., $v_{ji} = f_m(s_j, t_i)$.

A *weighted dataset* collects all the measurements for a measure $m \in M$ associated with any sensors in S for all the points of time in T. It is modeled as a set of pairs (i) point of time and (ii) a set of weighted items. Each set of weighted items will be hereafter denoted as *transaction*. An example of weighted dataset for the temperature measure is shown in Table 1.

Definition 2. *WSN weighted dataset.* Let S be the set of sensors in a WSN, m be a measure in M, T be a set of points of time, and V be a set of measurements.

A WSN weighted dataset $\mathcal{D}$ is a set of pairs (t_i, tr_i) where $t_i \in T$ and $tr_i = \{\langle s_j, v_{ji} \rangle | s_j \in S\}$ is a set of weighted items. For $\mathcal{D}$ the following properties hold. (i) Every $s_j \in S$ is an item in $\mathcal{D}$. (ii) Each transaction $tr_i \in \mathcal{D}$ is associated with a unique sampling time instant t_i.

The dataset in Table 1 reports the WSN weighted dataset. It collects the temperature measurements (sensor readings) sampled at 6 time instants by the example WSN in Fig. 1. At time t_1 sensors s_1 and s_2 read values 15 °C and 27 °C, respectively. These values represent the weights of sensors s_1 and s_2 at time t_1. Sensors s_1 and s_2 are items, while pairs $\langle s_1, 15°C \rangle$ and $\langle s_2, 27°C \rangle$ are weighted items.

While monitoring a measure over a large time interval, a few sensor readings could be missing. To replace missing data values in WSNs, Various approaches have been proposed [16]. For each sensor, WSNA replaces its missing values by taking the average of the last value before and of the first value after the gap. However, more sophisticated strategies can be easily integrated as well.

3.2. Unexpected pattern mining

We mine infrequent itemsets from the WSN weighted dataset to discover the combinations of WSN sensors showing unexpected behaviors. To this purpose, we applied the MIWI mining [10]. The fundation behind itemset mining techniques and the motivations behind the choice of the MIWI mining algorithm are summarized below.

Traditional itemset mining approaches (e.g., Apriori [4], FP-Growth [19]) cannot be applied to WSN data, because all the data items are assumed to be equally relevant within the analyzed data. Conversely, in the context of WSNs, items

Table 1
Example of WSN weighted dataset for the temperature measure.

Time stamp	Sensor readings
t_1	$\langle s_1, 15\,°C\rangle, \langle s_2, 27\,°C\rangle, \langle s_3, 20\,°C\rangle, \langle s_4, 22\,°C\rangle$
t_2	$\langle s_1, 15\,°C\rangle, \langle s_2, 18\,°C\rangle, \langle s_3, 16\,°C\rangle, \langle s_4, 22\,°C\rangle$
t_3	$\langle s_1, 18\,°C\rangle, \langle s_2, 15\,°C\rangle, \langle s_3, 18\,°C\rangle, \langle s_4, 18\,°C\rangle$
t_4	$\langle s_1, 27\,°C\rangle, \langle s_2, 15\,°C\rangle, \langle s_3, 18\,°C\rangle, \langle s_4, 27\,°C\rangle$
t_5	$\langle s_1, 24\,°C\rangle, \langle s_2, 22\,°C\rangle, \langle s_3, 15\,°C\rangle, \langle s_4, 22\,°C\rangle$
t_6	$\langle s_1, 20\,°C\rangle, \langle s_2, 22\,°C\rangle, \langle s_3, 15\,°C\rangle, \langle s_4, 22\,°C\rangle$

Table 2
Patterns mined from the dataset in Table 1. $\xi = 19\,°C$.

Pattern	IWI-support	MIWI/ Not MIWI	Pattern	IWI-support	MIWI/ Not MIWI
$\{s_3\}$	17 °C	MIWI	$\{s_1, s_2, s_4\}$	17 °C	Not MIWI
$\{s_1, s_2\}$	17 °C	MIWI	$\{s_2, s_3, s_4\}$	16 °C	Not MIWI
$\{s_2, s_4\}$	19 °C	MIWI	$\{s_1, s_2, s_3\}$	15 °C	Not MIWI
$\{s_3, s_4\}$	17 °C	Not MIWI	$\{s_1, s_3, s_4\}$	15 °C	Not MIWI
$\{s_1, s_3\}$	16 °C	Not MIWI	$\{s_1, s_2, s_3, s_4\}$	15 °C	Not MIWI
$\{s_2, s_3\}$	16 °C	Not MIWI			

(i.e., sensors) are characterized by different weights (sensor measurements) at different sampling times. The problem of extracting itemsets from weighted data is known as the *weighted itemset mining problem* [39]. A *weighted itemset* I is a set of k distinct items occurring in a weighted dataset [39]. In the context of WSNs, a weighted itemset is a set of k sensors in the WSN dataset $\mathcal{D}$.

Example. $\{s_1\}$, $\{s_2\}$, and $\{s_1, s_2\}$ are examples of weighted itemsets mined from the dataset in Table 1.

The notation used above is in compliance with the previous works on weighted itemset mining (e.g., [39]). In our context, item weights occurring in the WSN dataset are used to drive the weighted itemset mining process. Hence, the mined itemsets are denoted as *weighted*, even if they do not include weights. For the sake of clarity, hereafter weighted itemsets will be simply denoted as itemsets whenever it is clear from the context.

The traditional (not weighted) itemset mining problem is commonly driven by well-known itemset quality measures. For example, the support of an itemset in a transactional dataset is the percentage of dataset transactions containing it [3]. For our purposes, the itemset support measure is extended, similar to [39], to the case of weighted data. Specifically, weighted itemset extraction is driven by the IWI-support measure. The IWI-support was introduced in [10] to efficiently address the Infrequent Weighted Itemset (IWI) mining problem.

Let I be a weighted itemset (i.e., a set of sensors) in $\mathcal{D}$. The *IWI-support* of an itemset I in the WSN dataset $\mathcal{D}$ is a weighted frequency of occurrence of I in $\mathcal{D}$. To weigh the occurrence of an itemset I in an arbitrary transaction $tr_i \in \mathcal{D}$, an aggregation function f (e.g., min, max, average, mode) is used to combine the weights of the I's items in tr_i. A more formal definition follows.

Definition 3. *IWI-support.* Let $\mathcal{D}$ be a WSN weighted dataset, let I be a weighted itemset in $\mathcal{D}$, let $MS_{fin}(\mathbb{R})$ be the set of all finite multisets over $\mathbb{R}$ and let M_{I,tr_i} be the finite multi-set of measurements of the sensors in I associated with transaction $tr_i \in \mathcal{D}$. Let $f : MS_{fin}(\mathbb{R}) \rightarrow \mathbb{R}$ be a function (e.g., minimum, maximum, average) that aggregates the sensor measurements.

The IWI-support of I in $\mathcal{D}$ is given by

$$\text{IWI-support}(I, \mathcal{D}) = \frac{\sum_{tr_i \in \mathcal{D}} f(M_{I,tr_i})}{|\mathcal{D}|}$$

The choice of the aggregation function depends on the considered use cases. Hereafter, we will consider $f = \min$ (i.e., it takes the least measurement acquired by any sensor in I at time t_i), because the selected patterns are particularly useful for pinpointing unexpected behaviors in WSNs. The integration of different aggregation functions will be discussed in Section 3.2.3.

Example. The IWI-support of itemset $\{s_1, s_2\}$ in Table 1 is 17 °C. This value is the average of the least temperature measurements acquired by either s_1 or s_2 at each sampling time instant from t_1 to t_6. For example, at time t_1 sensor s_1 measured the least temperature (15 °C against 27 °C measured by s_2). The weight contributions to the IWI-support of $\{s_1, s_2\}$ are 15 °C at times $t_1 - t_4$, 22 °C at time t_5, and 20 °C at time t_6.

Given a WSN dataset $\mathcal{D}$ and an IWI-support threshold ξ, let us denote as *Infrequent Weighted Itemsets* (IWIs) the weighted itemsets whose IWI-support$(I, \mathcal{D}) \leq \xi$. Table 2 reports the set of IWIs extracted from the WSN dataset in Table 1 by enforcing $\xi = 19\,°C$. In the performed experiments, ξ was set to values between 19°C and 21°C, because these values were assumed to be the minimum temperature of livable habitats [12]. Each IWI represents a combination of sensors for which at least one sensor reading is, on average, less than or equal to ξ at every sampling time instant. With convenient abuse of notation, hereafter this situation will be also denoted as *correlation* between sensor readings. For example, $\{s_1, s_2\}$ indicates that, on average, either sensor s_1 or s_2 (or both of them) have measured a temperature less than or equal to 19 °C for every sampling time instant (possibly in an alternate fashion).

Experts are commonly interested only in the minimal combinations of sensors representing anomalous situations. WSNA selects a worthwhile IWI subset, namely the *Minimal IWIs* (MIWIs). MIWIs are IWIs of minimal size, i.e., IWIs for which none of their proper subsets is an IWI. MIWIs represent the minimal combinations of sensors for which at least one sensor reading is, on average, below the temperature threshold ($\xi = 19\,°\mathrm{C}$) for every sampling time instant.

Example. $\{s_1, s_2\}$ is a MIWI because neither s_1 nor s_2 satisfies the IWI-support threshold. Conversely, $\{s_1, s_2, s_3\}$ is a non-minimal IWI because at least one of its proper subsets (e.g., $\{s_1, s_2\}$) is an IWI. Column 3 of Table 2 differentiates between minimal and non-minimal IWIs.

While considering decreasing functions (e.g., $f = \min$) MIWI extraction can be efficiently accomplished thanks to the following property.

Lemma 1. *Let f be a decreasing aggregation function, i.e., given two multi-sets of item weights V_1 and V_2, such that $V_1 \subseteq V_2$, the following condition holds: $f(V_1) \geq f(V_2)$. Assume that the IWI-support measure of an itemset is computed using function f. The IWI-support values of itemsets X and Y, $X \subseteq Y$, based on function f satisfy the following property: IWI-support$(X, \mathcal{D}) \geq$ IWI-support$(Y, \mathcal{D})$.*

Proof. Let $\mathcal{D}$ be a weighted WSN dataset and let tr_i be an arbitrary transaction in $\mathcal{D}$. Let $\mathrm{V}(X, tr_i) = \{v_{ji} | \langle s_j, v_{ji} \rangle \in tr_i \wedge s_j \in X\}$ be the multi-subset of weights in tr_i associated with items in X. Since $X \subseteq Y$ then $\mathrm{V}(X, tr_i) \subseteq \mathrm{V}(Y, tr_i)$. Thus, applying the decreasing function f the following inequality holds: $f(\mathrm{V}(X, tr_i)) \geq f(\mathrm{V}(Y, tr_i))$. By summing up the values returned by function f over all the transactions in $\mathcal{D}$, we get $\sum_{tr_i \in \mathcal{D}} f(V(X, tr_i)) \geq \sum_{tr_i \in \mathcal{D}} f(V(Y, tr_i))$. Therefore, if $X \subseteq Y$ then $\text{IWI-support}(X, \mathcal{D}) \geq \text{IWI-support}(Y, \mathcal{D})$. □

If an IWI I is infrequent, then all of its supersets can be deemed as not useful for manual inspection by domain experts, because they do not provide any additional information.

3.2.1. Distance-based constraints

While analyzing WSN measurements experts may be interested in analyzing only the measurements acquired by (a) nearby sensors, because readings acquired in the same environment are most likely to be correlated with each other, or (b) distant sensors, to correlate sensor measurements acquired in different environments or under different conditions.

Experts may consider the spatial distance between the sensors in the WSN topology to drive the exploration of the mined patterns. Specifically, WSNA allows experts to constrain the minimum or maximum distance (computed on the WSN topology) between each couple of sensors in the selected MIWIs, according to the use case of interest. Given a distance threshold δ, WSNA extracts (a) the MIWIs satisfying the *closeness constraint*, i.e., the combinations of nearby sensors such that the spatial distance between each couple of sensors is *no greater than* δ or (b) the MIWIs satisfying the *distance constraint*, i.e., the combinations of distant sensors such that the spatial distance between each couple of sensors is *above* δ.

Definition 4. *MIWI selection criteria.* Let $\mathcal{D}$ be a WSN weighted dataset, S the set of sensors in $\mathcal{D}$, and $\mathrm{d}(s_i, s_j)$ the distance between two arbitrary sensors $s_i, s_j \in S \mid i \neq j$, in the WSN topology. Let δ be a sensor distance threshold. MIWIs are selected according to one of the following constraints:

(a) A MIWI I in $\mathcal{D}$ satisfies the *closeness constraint* iff $\forall\, s_i, s_j \in I$, $\mathrm{d}(s_i, s_j) \leq \delta$
(b) A MIWI I in $\mathcal{D}$ satisfies the *distance constraint* iff $\forall\, s_i, s_j \in I$, $\mathrm{d}(s_i, s_j) > \delta$

In the performed experiments, $d(s_i, s_j)$ was computed as the Euclidean distance between s_i and s_j [20]. Notice that a sensor may occur in several MIWIs (see Definition 4).

Example. let us consider the MIWIs $\{s_1, s_2\}$ and $\{s_2, s_4\}$ extracted from the WSN dataset in Table 1. Let us assume that the maximum distance threshold δ is set to 2 m (2 m), which implies that the selected combinations of sensors are likely to be placed within the same environment. $\{s_1, s_2\}$ satisfies the closeness constraint because $d(s_1, s_2) \leq$ 2m. A complementary analysis could prompt experts to consider only the correlations between relatively distant sensors. Since $d(s_2, s_4) > 2\,\mathrm{m}$, MIWI $\{s_2, s_4\}$ satisfies the distance constraint.

3.2.2. The algorithm

Given a WSN weighted dataset $\mathcal{D}$, an IWI-support threshold ξ, and a distance-based constraint C, we extract all the MIWIs from $\mathcal{D}$ satisfying both ξ and C. To accomplish this task, we exploit a newly proposed algorithm, i.e., the Tuned MIWI Miner algorithm. The Tuned MIWI Miner algorithm extends the MIWI Miner algorithm [10] in the following directions.

(i) *Extraction of a more compact set of patterns.* MIWI Miner extracts all the possible MIWIs, whereas Tuned MIWI Miner mines only the subset of MIWIs satisfying a distance-based constraint (see Section 3.2.1).

(ii) *Pushing constraint into the knowledge extraction process.* Tuned MIWI Miner pushes the distance-based constraints deep into the mining process, whereas MIWI Miner would need an ad hoc postpruning phase to select the subset of MIWIs of interest.

As shown in Section 4, pushing the distance-based constraints into the MIWI mining process significantly limits the number of mined MIWIs. Thus, the output pattern set is more compact and manageable by domain experts.

The pseudo-code of the Tuned MIWI Miner is reported in Algorithms 1 and 2. The algorithm relies on three main steps:

Algorithm 1 Tuned MIWI-Miner(T, ξ, C).

```
Input:  $T$, a weighted transactional dataset
Input:  $\xi$, a maximum IWI-support threshold
Input:  $C$, a distance-based constraint
Output: $\mathcal{F}$, the set of MIWIs satisfying $\xi$
1: $\mathcal{F} \leftarrow \emptyset$ /* Initialization *//* Scan $T$ and build its the equivalent transaction set*/
2: $TE \leftarrow$ equivalentTransactionSet($T$)/* Create the initial FP-tree from $TE$ */
3: $Tree \leftarrow$ FP-tree($TE$)
4: $\mathcal{F} \leftarrow$ Tuned-MIWIMining($Tree$, $\xi$, $C$, $\emptyset$) /* Recursive mining function*/
5: return $\mathcal{F}$
```

Algorithm 2 Tuned-MIWIMining($Tree$, ξ, C, px).

```
Input:  $Tree$, a FP-tree
Input:  $\xi$, a maximum IWI-support threshold
Input:  $C$, a distance-based constraint
Input:  $px$, the set of items/projection patterns with respect to which $Tree$ has been generated
Output: $\mathcal{F}$, the set of IWIs extending $px$
1: $\mathcal{F} \leftarrow \emptyset$
2: for all item $i_j$ in the header table of $Tree$ $|\forall i_q \in px, distance(i_j, i_q)$ satisfies the distance-based constraint $C$ do
3:     $I \leftarrow px \cup \{i_j\}$ /* Generate a new potential infrequent itemset $I$ by joining $px$ and $i_j$ *//* If $I$ is infrequent store it */
4:     if IWI-support($I$) $\leq \xi$ then
5:         $\mathcal{F} \leftarrow \mathcal{F} \cup \{I\}$
6:     end if/* Build $I$'s conditional FP-tree */
7:     $Tree_I \leftarrow$ createFP-tree($Tree$, $I$)
8:     if $Tree_I \neq \emptyset$ then
9:         $\mathcal{F} \leftarrow \mathcal{F} \cup$ Tuned-MIWIMining($Tree_I$, $\xi$, $C$, $I$) /* Recursive mining*/
10:    end if
11: end for
12: return $\mathcal{F}$
```

(a) Creation of an equivalent transaction set, in which weights are uniformly distributed within each equivalent transaction (Algorithm 1, line 2),

(b) Creation of a compact in memory representation of the transactional dataset based on an FP-tree-like structure (Algorithm 1, line 3), and

(c) Recursive itemset mining from the FP-tree-like structure (Algorithm 1, line 4).

Steps (a) and (b) rely on the functionalities provided by the baseline algorithm version (MIWI Miner [10]), while step (c) is peculiar to the extended version. Specifically, at step (a) the *equivalentTransactionSet* function is used to transform the initial dataset, in which each transaction may contain items with different weights, in an equivalent dataset where all the items of the same transaction have the same weight. This transformation is exploited to efficiently mine infrequent itemsets. Specifically, each weighted transaction corresponds to an equivalent weighted transaction set. Item weights in the original transaction are spread, based on their relative significance, among the corresponding equivalent transactions. While using the minimum weighting function, the equivalence procedure first considers the least weight occurring in the original transaction as current reference weight and generates an equivalent transaction of equally weighted items. Next, an iterative procedure only considers, for the subsequent steps, the set of items S contained in the original transaction and having weight strictly higher than the reference weight. Items in S are combined in a new equivalent transaction. At this stage, the new value of reference weight is set to the minimum weight among the items in S reduced by the previous reference weight value. Next, set S is further pruned by excluding items with the current reference weight once more. The above procedure is iterated until S is empty. The proposed transformation is particularly suitable for compactly representing the original dataset by means of an FP-tree index [19]. A more detailed description of the above-mentioned data transformation procedure is given in [10].

Once the equivalent set has been generated, at step (b) it is stored in main memory by using a prefix-tree data structure. Specifically, an FP-tree is created [19]. Finally, (c) the recursive Tuned-MIWIMining algorithm is invoked on the generated FP-tree (Algorithm 1, line 4). In the recursive itemset mining step, the newly proposed distance-base constraint is enforced. Specifically, Tuned MIWI Miner adopts a different item pruning strategy with respect to MIWI Miner [10]. To generate a potentially new MIWI I, prefix px is extended with an item i_j (Algorithm 2, line 3). To prevent the generation of the MIWIs that do not satisfy the distance-based constraint, the aforesaid extension is performed only for the items i_j that satisfy the distance-based constraint C with respect to all the items $i_q \in px$ (Algorithm 2, line 2).

Items satisfying the distance constraint are considered one at a time and used to extend the current prefix p_x and to generate new candidate itemsets I. If I is infrequent, then it is included in the set of infrequent itemsets (Algorithm 2, line 5). To generate further extensions of the current candidate itemset, a conditional FP-tree, containing only the transactions containing the items in I, is created by using the same createFP-tree function described in [10] (Algorithm 2, line 7). Finally, the recursive Tuned MIWI Mining function is invoked on the new conditional FP-tree (Algorithm 2, line 9).

3.2.2.1. Complexity analysis. Tuned MIWI Miner is an FP-growth-like mining algorithm with an embedded distance-based constraint. Similar to FP-growth [19], the complexity of Tuned MIWI Miner is linear with respect to the number of mined itemsets, which is combinatorial with the number of items ($2^{\#items}$ in the worse case) [19], if the distance-based constraint is not enforced. However, thanks to the distance-based constraint, which has been pushed deep into the itemset mining process (see Algorithm 2) the actual number of explored attribute combinations is significantly lower.

The Tuned MIWI Miner algorithm is *complete* (i.e., it extracts all the MIWIs satisfying the IWI-support and the distance-based constraint). Tuned MIWI Miner is *not correct*, because, similar to other FP-Growth-like itemset mining algorithms (e.g., [28]), it potentially generates a superset of the patterns satisfying the constraints. Below, we report a proof of completeness and a counterexample showing the non-correctness of the Tuned MIWI Miner algorithm.

3.2.2.2. Proof of completeness. By contradiction, let us suppose that an itemset satisfying the given constraint is not extracted. Since the FP-Growth-like itemset extraction is complete [19], then the distance-based constraint strategy wrongly prunes the candidate itemset. According to Definition 4, itemset I is pruned if it contains any pair of items not satisfying the constraint. If function $f = \min$ then, thanks to Lemma 1, all the extensions I_1, I_2 of I ($I \subset I_1$, I_2) are pruned because, by construction, they do not satisfy the constraint as well. Contradiction.□

3.2.2.3. Non-correctness: a counterexample. Similar to most FP-Growth-like itemset mining algorithms (e.g., [28]), the Tuned MIWI Miner algorithm may generate a superset of the non-minimal itemsets. According to the algorithm described in Algorithm 2, projection trees are recursively generated and visited. However, the order in which items occur in the header table matters.

Let us consider the triplet of items *a, b*, and *c*. Let us suppose that itemsets {*a, c*} and {*a, b, c*} are both infrequent with respect to the IWI-support threshold ξ, whereas all the other combinations of the aforesaid items are frequent. Hence, {*a, c*} is a MIWI, whereas {*a, b, c*} is not. While generating the projection tree associated with item *a* the order of appearance of items *b* and *c* matters. Specifically, if the two items have the same local IWI-support value, item *b* is considered first, because it precedes *c* in alphabetical order. In this case, itemset {*a, b, c*} is generated even if it is not minimal.

The presence of redundant (non-minimal) itemsets slightly affects the quality of the mining result. For example, in the experiments performed on synthetic data less than 5% of the mined itemsets were non-minimal, while redundant itemsets were not extracted at all from real data.

3.2.3. Use of different aggregation functions

The use of aggregation functions *f* other than *min* could enable the application of the proposed approach in different use case scenarios. Tuned MIWI Miner supports of aggregation measures *f* other than *min* provided that a splitting procedure is applied to the raw WSN readings prior to MIWI mining. Specifically, each transaction of the original WSN weighted dataset has to be replaced with an equivalent transaction set corresponding to all the possible item subsets. In other words, for every transaction $tr_i \in \mathcal{D}$ consisting of N sensor readings, the splitting procedure generates up to 2^N transactions, each one corresponding to a distinct subset of items in tr_i. This procedure potentially generates Big datasets, because the cardinality of the WSN weighted datasets combinatorially increases. To guarantee the scalability of the mining process towards Big datasets, the proposed centralized solution can be extended towards a distributed architecture. Perspectives of extension of the current architecture are given in Section 5.

4. Experimental results

The efficiency and effectiveness of the proposed approach were evaluated on real and synthetic data.

Real WSN data. We considered two real WSN datasets related to different case studies. The first dataset, named *Heat*, concerns heating system monitoring. It relates a WSN deployed in April 2009 within a campus network. The WSN consists of a network of autonomous sensors that were placed in different research laboratories in the same building and floor as well as in their adjacent corridor. The WSN topology is depicted in Fig. 3. The campus WSN consists of 16 nodes. Each node consists of a Tmote Sky module, which is a low power wireless module for use in sensor networks. It features an IEEE 802.15.4 wireless transceiver with antenna, a USB connection, and integrates humidity, light, and temperature sensors. All the nodes were configured to transmit their daily measures in bulk at a certain time of the day. Specifically, antennas were switched on for half hour at 5.30 pm each day. The half-hour period allowed us to avoid transmission losses due to drift in the mote clocks. The bulk transmission were chosen instead of the real-time monitoring because it greatly improved the mote lifetime, which was, on average, up to a month.

The second dataset, named *Pollution*, concerns the analysis of the impact of polluting agents on environmental conditions in a urban scenario. It is an open dataset provided by ARPA Lombardia, an organization devoted to the protection of

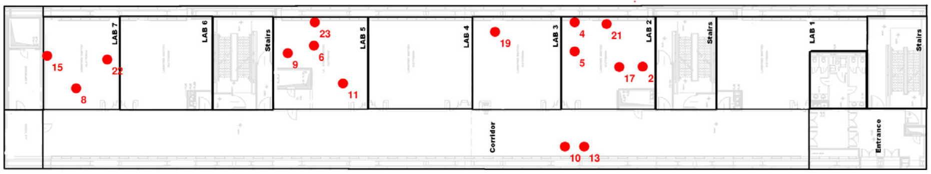


Fig. 3. WSN topology associated with the *Heat* dataset

Table 3
Combinations of nearby sensors (Weekdays. Working hours. Maximum distance threshold $\delta = 8$ m) .

Average temperature threshold ξ (°C)	Patterns (IWI-sup)	Average temperature measured by each sensor (°C)
20.6	$\{s_{10}\}$ (17.64)	
	$\{s_8, s_{22}\}$ (20.54)	$\{s_8\}$ (22.31) – $\{s_{22}\}$ (20.63)
	$\{s_{15}\}$ (20.34)	
21.8	$\{s_6, s_9\}$ (21.73)	$\{s_6\}$ (21.97) – $\{s_9\}$ (21.94)
	$\{s_{10}\}$ (17.64)	
	$\{s_{13}\}$ (21.61)	
	$\{s_{15}\}$ (20.34)	
	$\{s_{22}\}$ (20.63)	
22.2	$\{s_6\}$ (21.97)	
	$\{s_9\}$ (21.94)	
	$\{s_{10}\}$ (17.64)	
	$\{s_{11}, s_{23}\}$ (22.04)	$\{s_{11}\}$ (22.23) – $\{s_{23}\}$ (22.44)
	$\{s_{13}\}$ (21.61)	
	$\{s_{15}\}$ (20.34)	
	$\{s_{22}\}$ (20.63)	

the environment. Pollutant concentrations are gathered by ARPA Lombardia through some monitoring stations located in the Italian Region Lombardia. Each station is equipped with a set of sensors, each one measuring the levels of a given pollutant. The provided data consist of a set of hourly or daily readings, depending on the type of pollutant. Each reading is characterized by the monitoring station identifier, the sensor identifier, the name of the measured pollutant, the concentration of the pollutant, and the date and hour of the reading. The dataset considered in this study collects the percentage level of Nitrogen dioxide (NO_2) acquired by different sensors located all over the city of Milan (Italy) on each day of 2013.

Synthetic data. To test the scalability of our approach, we used the public dataset generator described in [10].

All the experiments were performed on a 3.0 GHz Intel Xeon system with 16 GB RAM, running Ubuntu 12.04 LTS.

4.1. Unexpected behavior discovery

The section analyzes the unexpected patterns mined in two different case studies and compares the information provided by the mined patterns with that provided by the analysis of individual sensor's readings.

4.1.1. Heating system monitoring

To detect possible malfunction of the heating system in the campus network, we analyzed the historical temperature readings acquired by the WSN. Temperature measurements in the *Heat* dataset were acquired every 15 min. over a time period of one month.

Let us analyze first the correlations between sensor readings acquired over the weekdays during the time slot from 9 A.M. to 6 P.M. To analyze the correlations between the readings of sensors placed within the same laboratory/in different laboratories, we set the maximum/minimum distance threshold δ to 8 m (i.e., a rough estimate of the lab size). Following the recommendations of the technical staff of the campus network, we considered 21 °C as a reliable estimate of the minimum temperature of a livable habitat. We assume that average temperatures below the aforesaid value are supposed to be critical [12]. In Tables 3 and 4 the itemsets mined by enforcing three representative temperature thresholds ξ close to the critical value are reported. Tables 3 and 4 compare also the average temperature associated with each pattern with the average reading values associated with each sensor in the pattern. As discussed below, the information provided by individual sensors is not sufficient to infer the same knowledge provided by unexpected patterns.

For example, pattern $\{s_8, s_{22}\}$ is an unexpected pattern representing a combination of two nearby sensors extracted by setting ξ to 20.6 °C. Both sensors s_8 and s_{22} are located in Lab7 (see Fig. 3). It indicates that, on average, at least one of the two sensors measured a temperature below the threshold during the considered time slot (see Column 2 in Table 3). This pattern may highlight a malfunction of the heating system in Lab7. This pattern is unexpected because considering the

Table 4
Combinations of distant sensors (Weekdays. Working hours. Minimum distance threshold $\delta = 8$ m).

Average temperature threshold ξ (°C)	Unexpected patterns (IWI-sup)	Avg. temperature measured by each sensor (°C)
20.6	$\{s_6, s_{22}\}$ (20.48)	$\{s_6\}$ (21.97) – $\{s_{22}\}$ (20.63)
	$\{s_6, s_{13}\}$ (20.59)	$\{s_6\}$ (21.97) – $\{s_{13}\}$ (21.61)
	$\{s_8, s_{13}\}$ (20.58)	$\{s_8\}$ (22.31) – $\{s_{13}\}$ (21.61)
	$\{s_9, s_{22}\}$ (20.58)	$\{s_9\}$ (21.94) – $\{s_{22}\}$ (20.63)
	$\{s_9, s_{13}\}$ (20.53)	$\{s_9\}$ (21.94) – $\{s_{13}\}$ (21.61)
	$\{s_{10}\}$ (17.64)	
	$\{s_{11}, s_{13}, s_{19}\}$ (20.59)	$\{s_{11}\}$ (22.23) – $\{s_{13}\}$ (21.61) – $\{s_{19}\}$ (23.03)
	$\{s_{13}, s_{22}\}$ (19.77)	$\{s_{13}\}$ (21.61) – $\{s_{22}\}$ (20.63)
	$\{s_{15}\}$ (20.34)	
	$\{s_{22}, s_{23}\}$ (20.56)	$\{s_{22}\}$ (20.63) – $\{s_{23}\}$ (22.44)
21.8	$\{s_6, s_8\}$ (21.75)	$\{s_6\}$ (21.97) – $\{s_8\}$ (22.31)
	$\{s_8, s_9\}$ (21.67)	$\{s_8\}$ (22.31) – $\{s_9\}$ (21.94)
	$\{s_8, s_{11}, s_{19}\}$ (21.72)	$\{s_8\}$ (22.31)– $\{s_{11}\}$ (22.23) – $\{s_{19}\}$ (23.03)
	$\{s_8, s_{19}, s_{23}\}$ (21.75)	$\{s_8\}$ (22.31)– $\{s_{19}\}$ (23.03) – $\{s_{23}\}$ (22.44)
	$\{s_{10}\}$ (17.64)	
	$\{s_{13}\}$ (21.61)	
	$\{s_{15}\}$ (20.34)	
	$\{s_{22}\}$ (20.63)	
22.2	$\{s_2, s_8\}$ (22.18)	$\{s_2\}$ (24.22) – $\{s_8\}$ (22.31)
	$\{s_6\}$ (21.97)	
	$\{s_8, s_{11}\}$ (21.88)	$\{s_8\}$ (22.31) – $\{s_{11}\}$ (22.23)
	$\{s_8, s_{17}\}$ (22.20)	$\{s_8\}$ (22.31) – $\{s_{17}\}$ (24.31)
	$\{s_8, s_{19}\}$ (21.96)	$\{s_8\}$ (22.31) – $\{s_{19}\}$ (22.44)
	$\{s_8, s_{23}\}$ (21.96)	$\{s_8\}$ (22.31) – $\{s_{23}\}$ (22.44)
	$\{s_9\}$ (21.94)	
	$\{s_{10}\}$ (17.64)	
	$\{s_{11}, s_{19}\}$ (22.06)	$\{s_{11}\}$ (22.23) – $\{s_{19}\}$ (22.44)
	$\{s_{13}\}$ (21.62)	
	$\{s_{15}\}$ (20.35)	
	$\{s_{19}, s_{23}\}$ (22.17)	$\{s_{19}\}$ (22.44) – $\{s_{23}\}$ (22.44)
	$\{s_{22}\}$ (20.63)	

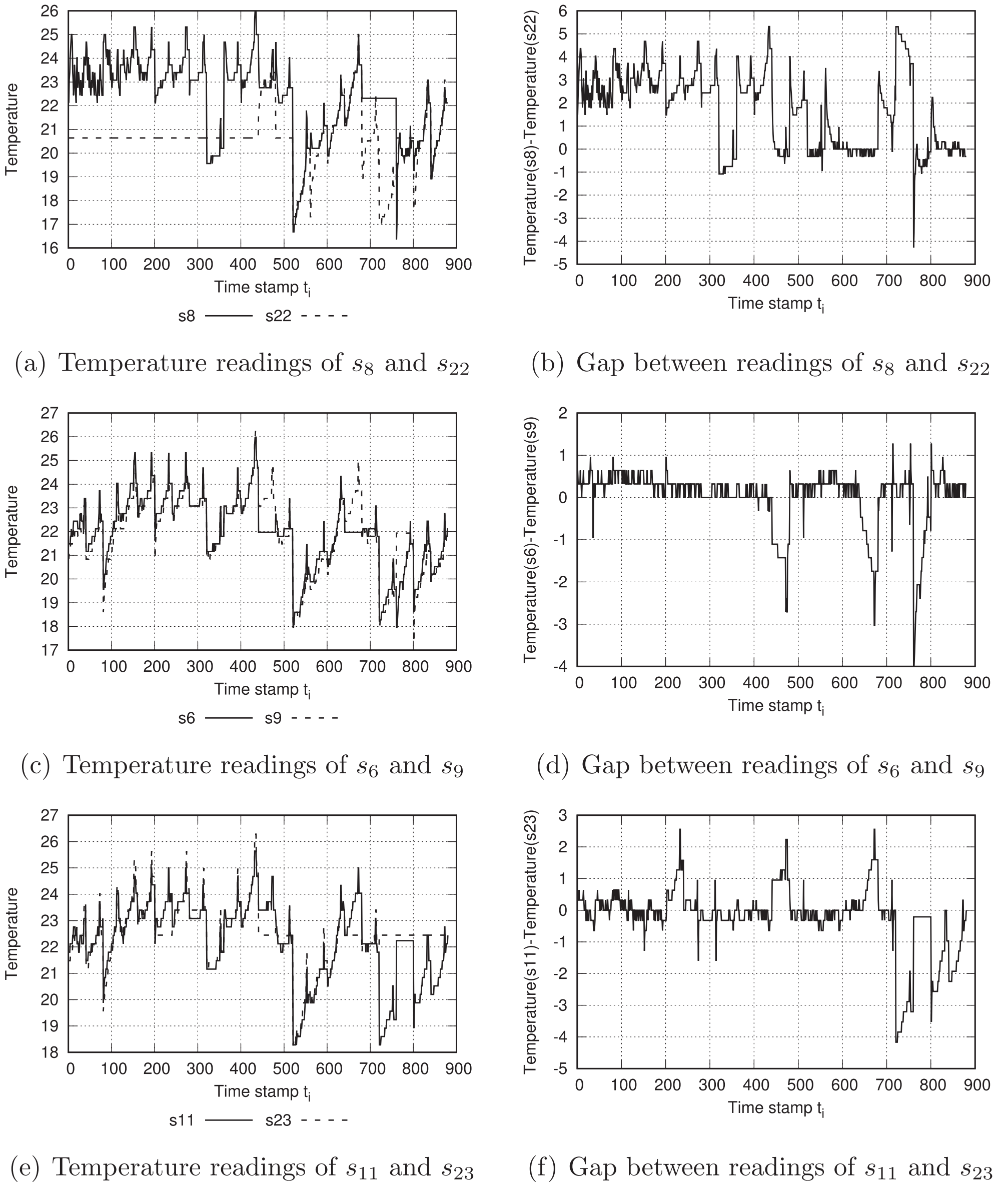
two sensors separately both of them measured an average temperature above the threshold (see Column 3). Hence, the two sensors measured low temperature values in an alternate fashion.

Experts may further investigate the correlation between the subsets of sensors identified by the unexpected patterns using their domain-specific knowledge. The relevance of the discovered patterns could be validated according to physical models, such as the model representing the distribution of air in a room [6]. For instance, according to the model of imperfectly mixed ventilated rooms, slight temperature variations in the same room (approximately 1 °C) can be deemed as acceptable. Therefore, by comparing the measurements of all the sensors in the same pattern, pattern can be classified as normal or critical. Fig. 4(a) and 4(b) plot the measurements acquired by sensors s_8 and s_{22} and the corresponding temperature gap, respectively. Based on these results, low temperatures were measured by the two sensors in an alternate fashion (the temperature gap reached 4 °C). Imbalances in temperature readings can be due, for example, to the presence of faulty radiators, which give off heat in a suboptimal manner. A similar validation process can be perform on each extracted pattern. According to the heat transfer model [6], experts may classify all the unexpected patterns as critical or normal. For example, Fig. 4(c) and 4(f) show similar trends associated with patterns $\{s_6, s_9\}$ and $\{s_{11}, s_{23}\}$. These patterns were extracted by setting ξ to 21.8 °C and 22.2 °C, respectively. Even in these cases, the temperature variations are above 4 °C at certain time instants, even if significant variations rarely occur.

A complementary analysis may focus on distant sensors, i.e., sensors located in different labs. Since all labs were supplied by the same heating system, analyzing this type of patterns can be useful for supporting heating system maintenance.

For example, pattern $\{s_{22}, s_{23}\}$ represents a correlation between the readings of sensors s_{22} and s_{23}, located in Lab7 and Lab5, respectively (see Table 4). Experts suppose that while radiators in Lab5 were giving off an excessive amount of heat, the supply for the other labs on average dropped. To back up this conclusion, we analyzed the average temperatures read by each sensor. The large gap between the average temperature readings of sensors s_{22} and s_{23} (i.e., 22.44 °C for sensor s_{23} in Lab5 against the 20.63 °C for sensor s_{22} in Lab7) confirms the soundness of their hypothesis.

Finally, we analyzed also the patterns mined over the weekend with different configuration settings (due to space constraints, the detailed results have been omitted). Since over the weekend the heating system has commonly turned off, no unexpected behavior appeared.



(a) Temperature readings of s_8 and s_{22}

(b) Gap between readings of s_8 and s_{22}

(c) Temperature readings of s_6 and s_9

(d) Gap between readings of s_6 and s_9

(e) Temperature readings of s_{11} and s_{23}

(f) Gap between readings of s_{11} and s_{23}

Fig. 4. Comparison between the readings of the sensors appearing in the same pattern. Weekdays. Working hours.

Spatial stratification of sensors. To mine unexpected patterns, we considered the measurements acquired from all the sensors located in each room. Therefore, we assumed a spatial stratification of the sensors within each room. Within each lab we empirically tested the hypothesis of spatially stratified heterogeneity [36] (i.e., the significance of sensor stratification) on the analyzed Wireless Sensor Network data by using the q-statistic method described in [37]. We made similar assumptions and tests for the corridor adjacent to the laboratories. Specifically, in [37] the authors proposed a method to measure the degree of spatial stratified heterogeneity and to test its significance. The q-statistic measures the correlation between the variances of population and per-stratum sampling. The q value is within [0,1] (0 if a spatial stratification of heterogeneity is not significant, and 1 if there is a perfect spatial stratification of heterogeneity). The probability density function F of the q-statistic is derived as a non-central chi-square function [37]. To test hypothesis of spatially stratified heterogeneity according to the procedure described in [37] we performed the following steps:

1. We computed the parameters of the F-distribution.
2. We estimated the critical value F_α at significance level $\alpha = 0.01$.

3. We quantitatively evaluated the size of the small area F^* of the F-distribution to the right of the critical value using the GeoDetector tool available at http://www.geodetector.org/ as well as the Web interface available at http://keisan.casio.com/exec/system/1180573166.
4. We tested the initial hypothesis of spatial stratified heterogeneity. If the area size F^* exceeds the critical value F_α (i.e., $F^* > F_\alpha$), then we accept the hypothesis, otherwise we reject it.

The achieved result confirmed the validity of the initial hypothesis for all the considered rooms ($F^*_{corridor} = 570.45 > F_{0.01} = 9.20$, $F^*_{Lab2} = 115.99 > F_{0.01} = 3.56$, $F^*_{Lab5} = 23.94 > F_{0.01} = 3.89$, $F^*_{Lab7} = 506.11 > F_{0.01} = 6.17$). Therefore, sensors within room represent spatially diversified temperature samples.

The resulting q values indicate the spatial stratification is almost perfect in Labs 2 and 5 (q equal to 0.024 and 0.095, respectively) and it is fair in Lab 7 and in the corridor (q equal to 0.277 and 0.245, respectively).

4.1.2. Analysis of the air pollution in a urban scenario

We used the historical WSN readings collected in the *Pollution* dataset to analyze the correlations between sensors measuring to level of air pollution in the city of Milan. Specifically, sensors acquire the level of Nitrogen dioxide (NO_2) in the air every 10 min for a time period of one year.

We analyzed the correlations between couples of nearby sensor readings by enforcing a maximum distance threshold equal to 5 km (i.e., a rough estimate of the diameter of a city area) and we set ξ to 50 μ g/m^3, because experts recommend this threshold value to discriminate between polluted areas and not. For example, the mined pattern $\{s_{10,279}, s_{5504}\}$ represents a combination of nearby sensors $s_{10,279}$ and s_{5504} placed 3.6 km far from each other. Since they are close to each other, they are expected to measure similar NO_2 values. However, this is not the case. While single sensors measured an averagely high pollution level, their combination is, on average, below the maximum threshold. Therefore, sensors $s_{10,279}$ and s_{5504} should be carefully monitored to detect possible causes of imbalances or malfunctioning.

4.2. Effect of the average temperature threshold

We analyzed the effect of the minimum IWI-support threshold on the characteristics of the patterns mined from the *Heat* dataset. Tables 3 summarizes the results achieved on the subset of WSN data acquired over the weekdays (from Monday to Friday; from 9 A.M. to 6 P.M.) by setting three representative temperature thresholds. While increasing the temperature threshold, a larger number of combinations become infrequent. Thus, the number of mined patterns increases. Although unexpected patterns may represent combinations of sensors of arbitrary size, even while setting relatively high threshold values the number of patterns discovered is still in the order of a few dozens for all the performed experiments. Conversely, decreasing the average temperature threshold results in a very compact and easy-to-read pattern set (e.g., only 3 patterns were extracted by enforcing $\xi = 20.6$ °C and $\delta = 8$ m).

4.3. Effect of the distance threshold

We also analyzed the effect of the distance threshold on the characteristics of the patterns mined from the *Heat* dataset. To perform our analyses, we considered the WSN data collected over the weekdays (from Monday to Friday; from 9 A.M. to 6 P.M.). Let us consider first the correlations between nearby sensor readings (i.e., closeness constraint). The distance threshold indicates for each mined pattern the maximal pairwise distance between its sensors. The number of mined patterns increases roughly linearly while increasing the distance threshold, because the pairs of nearby sensors are more likely to occur.

Let us consider now the correlations between distant sensor readings (i.e., distance constraint). The distance threshold indicates the minimal distance between each pair of sensors. The number of mined patterns is inversely proportional to the minimum distance threshold.

4.4. Scalability

This section analyzes the scalability of the proposed approach on synthetic data. We compared the performance of the WSNA system, in terms of execution time and number of mined patterns, with that of two baseline system versions, called PostMIWI and PostMINIT. Baseline systems perform (i) WSN data preparation, (ii) MIWI extraction driven by the support threshold, and (iii) Pattern selection based on postprocessing, to discard those patterns not satisfying the closeness/distance constraint. To extract unexpected patterns, PostMIWI relies on the MIWI Miner algorithm [10], whereas PostMINIT relies on the MINIT algorithm [18].

We compared the system execution times by varying the number of dataset transactions (i.e., the number of WSN readings). Specifically, we generated synthetic datasets with size ranging from 10,000 to 10,000,000 readings. Fig. 5 summarizes the results achieved by enforcing $\xi = 50$ and the closeness constraint ($\delta = 1$ m). Similar results were obtained with different values of ξ and δ. Both algorithms based on IWI Miner scale approximately linearly with the dataset cardinality. However, in all the performed tests, Tuned IWI Miner performs at least one order of magnitude better than PostMIWI. Unlike PostMIWI and Tuned IWI, PostMINIT was unable to process datasets with more than 100,000 transactions (i.e., readings). On the other

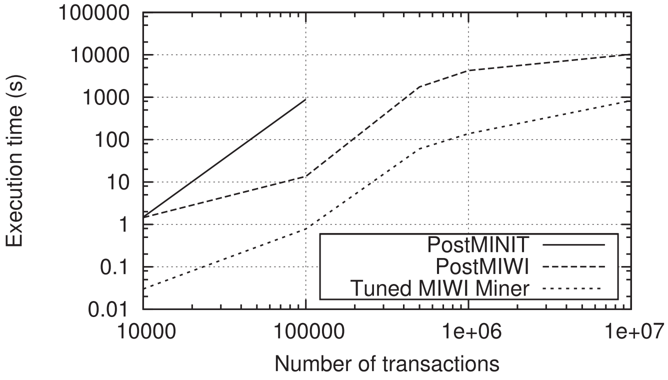


Fig. 5. Scalability with the number of transactions.

hand, thanks to the pushing of the distance-based constraints Tuned MIWI Miner achieved a significant time reduction w.r.t. MIWI Miner. Specifically, both PostMIWI and PostMINIT need an ad hoc postprocessing phase to filter out the uninteresting patterns, whereas Tuned MIWI Miner WSNA prevents their extraction. For most of the considered settings, the reduction in the number of generated MIWIs achieved by Tuned MIWI Miner with respect to MIWI Miner ranges between 95% and 99%.

5. WSNA: Perspectives of extension in a distributed environment

With the diffusion of smart cities, Wireless Sensor Networks (WSNs) have generated and collected readings at an unprecedented rate, to such an extent that WSN data rapidly scale towards “Big Data” [43]. Even when the number of network sensors is limited, collecting measurements for a long time at a high frequency produces huge data collections. Hence, analyzing WSN datasets often becomes computationally prohibitive. To efficiently analyze Big Data, a promising research direction is the study of distributed solutions. The extension of the Tuned MIWI Miner algorithm in a distributed environment entails the following steps, which can be performed by one or more MapReduce tasks running on an Hadoop cluster:

1. *Dataset sharding*. Each sensor periodically sends a reading file containing all the readings acquired over a given time period. The content of these files must be transformed to generate the transactional data representation of the readings. To achieve this goal, each node implements a mapper that receives as value the collected readings and generates a set of pairs (*key, value*), where *key* is a time stamp and *value* a pair (*sensor identifier, reading*). The reducer aggregates all the pairs with the same key (i.e., the same time stamp) and generates one transaction for each time stamp. Each transaction is a list of *N* pairs (*item, weight*), where *item* is a sensor identifier, *weight* is its corresponding reading, and *N* is the number of network sensors.

2. *Parallel counting*. This activity entails IWI-support counting for each item in the WSN dataset. This task is performed by means of a MapReduce job that exploits a word count-like solution to compute the complete set of items and their corresponding IWI-support values.

3. *Item clustering*. All the items in the dataset are split into disjoint groups. Since a WSN dataset typically contains a limited number of items (sensors), this step can be performed in main memory. Items are ranked by IWI-support and clustered by maximizing the similarity between their IWI-support values. Each item group identifies a distinct dataset portion on which the Tuned MIWI miner can be separately run. To reduce the computational overhead of the following steps more advanced strategies for item clustering can be also implemented (e.g., generating large/small clusters of frequent items or large clusters of infrequent items).

4. *Parallel MIWI mining*. This activity will be performed by means of a MapReduce job. The Mapper instance receives as input the groups of items generated at Step 3, processes the complete dataset one shard at a time, and generates for each group of items the corresponding set of projected transactions. The reducer instance receives as input the output of the mapper (i.e., a set of projected transactions for each group of items) and runs the Tuned MIWI miner algorithm (see Section 3.2.2) on the transaction set of each group. Each reducer instance outputs a disjoint subset of patterns.

5. *Partial result aggregation*. This job collects the patterns discovered at step 4 and it merges them to generate the complete set of interesting patterns.

6. Conclusions and future work

This paper presents a novel data mining approach to discovering unexpected behavior in Wireless Sensor Networks (WSNs). It focuses on supporting domain experts in the analysis of potentially large sets of past sensor readings. The experiments demonstrate the efficiency and applicability of the proposed approach. Real WSN data, acquired in different contexts, were analyzed and the achieved results were validated with the help of domain experts. Future extensions of this work entail the development of scalable services for WSN data monitoring (see Section 5) and the application of the proposed approach to data coming from diverse contexts (e.g., social networks, healthcare systems).

References

[1] A. Acquaviva, D. Apiletti, A. Attanasio, E. Baralis, F.B. Castagnetti, T. Cerquitelli, S. Chiusano, E. Macii, D. Martellacci, E. Patti, Enhancing energy awareness through the analysis of thermal energy consumption, in: Proceedings of the Workshop EDBT/ICDT, CEUR-WS, 2015.

[2] A. Acquaviva, D. Apiletti, A. Attanasio, E. Baralis, L. Bottaccioli, F.B. Castagnetti, T. Cerquitelli, S. Chiusano, E. Macii, D. Martellacci, E. Patti, Energy signature analysis: knowledge at your fingertips, in: Proceedings of the IEEE International Congress on Big Data, New York City, NY, USA, 2015, pp. 543–550.

[3] R. Agrawal, T. Imielinski, Swami, Mining association rules between sets of items in large databases, in: Proceedings of the ACM SIGMOD, 1993, pp. 207–216.

[4] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: J.B. Bocca, M. Jarke, C. Zaniolo (Eds.), Proceedings of the Twenty International Conference on Very Large Data Bases VLDB'94,, Morgan Kaufmann, 1994, pp. 487–499.

[5] D. Apiletti, E. Baralis, T. Cerquitelli, Energy-saving models for wireless sensor networks, Knowl. Inf. Syst. 28 (2011) 615–644.

[6] H.B. Awbi, Air Distribution in Rooms: Ventilation for Health and Sustainable Environment, Elsevier Science, 2000.

[7] A.M. Aziz, A new adaptive decentralized soft decision combining rule for distributed sensor systems with data fusion, Inf. Sci. (Ny) 256 (2014) 197–210. Business Intelligence in Risk Management.

[8] A. Boukerche, S. Samarah, In-network data reduction and coverage-based mechanisms for generating association rules in wireless sensor networks, IEEE Trans. Veh. Technol. 58 (2009) 4426–4438.

[9] L. Cagliero, T. Cerquitelli, S. Chiusano, P. Garza, X. Xiao, Predicting critical conditions in bicycle sharing systems, Computing 99 (2017) 39–57.

[10] L. Cagliero, P. Garza, Infrequent weighted itemset mining using frequent pattern growth, IEEE Trans. Knowl. Data Eng. 26 (2014) 903–915.

[11] T. Cerquitelli, E.D. Corso, Characterizing thermal energy consumption through exploratory data mining algorithms, in: Proceedings of the Workshops of the EDBT/ICDT Joint Conference, EDBT/ICDT Workshops, Bordeaux, France, 2016.

[12] C.M. Chiang, C.M. Lai, A study on the comprehensive indicator of indoor environment assessment for occupants's health in taiwan, Build Environ 37 (2002) 387–392.

[13] Y.C. Chung, I.F. Su, C. Lee, An efficient mechanism for processing similarity search queries in sensor networks, Inf. Sci. (Ny) 181 (2011) 284–307.

[14] C. David, L.N. D., L.T. Tsung-Te, P. Cong, M. Xiangying, G. Qing, L. Fan, Z. Feng, Balancing energy, latency and accuracy for mobile sensor data classification, in: Proceedings of the Ninth ACM Conference on Embedded Networked Sensor Systems, SenSys '11, ACM, New York, NY, USA, 2011, pp. 54–67.

[15] S. Goel, T. Imielinski, Prediction-based monitoring in sensor networks: taking lessons from MPEG, SIGCOMM Comput. Commun. Rev. 31 (2001) 82–98.

[16] L. Gruenwald, H. Yang, M.S. Sadik, R. Shukla, Using Data Mining to Handle Missing Data in multi-Hop Sensor Network Applications, mobiDE '10, ACM, New York, NY, USA, 2010, pp. 9–16.

[17] A. Gupta, A. Mittal, A. Bhattacharya, Minimally infrequent itemset mining using pattern-growth paradigm and residual trees, in: Proceedings of the COMAD, pp. 57–68.

[18] D.J. Haglin, A.M. Manning, On minimal infrequent itemset mining, in: Proceedings of the International Conference on Data Mining, DMIN'07, CSREA Press, 2007, pp. 141–147.

[19] J. Han, J. Pei, Y. Yin, Mining frequent patterns without candidate generation, in: Proceedings of the ACM SIGMOD International Conference on Management of Data, 2000, pp. 1–12.

[20] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd, Cambridge University Press, ISBN: 0521540518, 2004.

[21] Harvard, The Evolution of Decision Making: How Leading Organizations are Adopting a Data-Driven Culture, A report by Harvard Business Review Analytic Services., Harvard Business Review Analytic Services, 2012.

[22] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in: Proceedings of the Thirty-Third Hawaii International Conference on System Sciences-Volume 8 HICSS '00, IEEE Computer Society, Washington, DC, USA, 2000, p. 8020.

[23] J.Y. Huang, I.E. Liao, Y.F. Chung, K.T. Chen, Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining, Inf. Sci. (Ny) 231 (2013) 32–44. Data Mining for Information Security.

[24] D. Ilic, S. Karnouskos, M. Wilhelm, A comparative analysis of smart metering data aggregation performance, in: Proceedings of the Eleventh IEEE International Conference on Industrial Informatics (INDIN), 2013, pp. 434–439.

[25] S. Karnouskos, D. Ilic, P.G.D. Silva, Assessment of an enterprise energy service platform in a smart grid city pilot, in: Proceedings of the Eleventh IEEE International Conference on Industrial Informatics (INDIN), 2013, pp. 24–29.

[26] Y. Kotidis, Snapshot queries: towards data-centric sensor networks, in: Proceedings of the ICDE, 2005, pp. 131–142.

[27] W.K. Lai, C.S. Fan, L.Y. Lin, Arranging cluster sizes and transmission ranges for wireless sensor networks, Inf. Sci. (Ny) 183 (2012) 117–131.

[28] H. Li, Y. Wang, D. Zhang, M. Zhang, E.Y. Chang, Pfp: Parallel Fp-growth for Query Recommendation, in: Proceedings of the ACM Conference on Recommender Systems, RecSys '08, ACM, New York, NY, USA, 2008, pp. 107–114.

[29] A.M. Manning, D.J. Haglin, J.A. Keane, A recursive search algorithm for statistical disclosure assessment, Data Min. Knowl. Discov. 16 (2008) 165–196. Software downloaded from http://mavdisk.mnsu.edu/haglin.

[30] J. Matousek, Lectures on Discrete Geometry, Springer-Verlag, New York, Inc., SecaucSus, NJ, USA, 2002.

[31] M.M. Rashid, I. Gondal, J. Kamruzzaman, Mining associated sensor patterns for data stream of wireless sensor networks, in: Proceedings of the Eighth ACM Workshop on Performance Monitoring and Measurement of Heterogeneous Wireless and Wired Networks, PM2HW2N '13, ACM, New York, NY, USA, 2013, pp. 91–98.

[32] M. Ren, L. Guo, Mining recent approximate frequent items in wireless sensor networks, in: Proceedings of the FSKD (2), 2009, pp. 463–467.

[33] J.V. Rethy, H. Danneels, V.D. Smedt, W. Dehaene, G.G.E. Gielen, A Low-power and low-voltage BBPLL-based Sensor Interface in 130nm CMOS for wireless sensor networks, in: Proceedings of the Design, Automation and Test in Europe, DATE 13, Grenoble, France, 2013, pp. 1431–1435.

[34] S. Samarah, A. Boukerche, A.S. Habyalimana, Target association rules: a new behavioral patterns for point of coverage wireless sensor networks, IEEE Trans. Comput. 60 (2011) 879–889.

[35] W.Z. Song, M. Xu, D. De, D. Heo, J.H. Kim, B.S. Kim, Ecpc: toward preserving downtime data persistence in disruptive wireless sensor networks, ACM Trans. Sen. Netw. 11 (2014) 24:1–24:22.

[36] J.F. Wang, A. Stein, B.B. Gao, Y. Ge, A review of spatial sampling, Spat Stat 2 (2012) 1–14.

[37] J.F. Wang, T.L. Zhang, B.J. Fu, A measure of spatial stratified heterogeneity, Ecol. Indic. 67 (2016) 250–256.

[38] L. Wang, L.D. Xu, Z. Bi, Y. Xu, Data cleaning for RFID and WSN integration, IEEE Trans. Ind. Inf. 10 (2014) 408–418.

[39] W. Wang, J. Yang, P.S. Yu, Efficient mining of weighted association rules (WAR), in: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD'00, 2000, pp. 270–274.

[40] S. Yoon, C. Shahabi, The clustered aggregation (CAG) technique leveraging spatial and temporal correlations in wireless sensor networks, ACM Trans. Sen. Netw. 3 (1) (2007) Article No. 3.

[41] D. Yu, P. Nanda, L. Cao, X. He, TCTM: an evaluation framework for architecture design on wireless sensor networks, Int. J. Sen. Netw. 14 (2013) 168–177.

[42] C. Zhang, C. Wang, D. Li, X. Zhou, C. Gao, Unspecific event detection in wireless sensor networks, in: Proceedings of the International Conference on Communication Software and Networks, ICCSN '09., 2009, pp. 243–246.

[43] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: concepts, methodologies and applications, ACM Trans. Intell. Syst. Technol. 5 (2014) 38:1–38:55.