



HAL
open science

SANDMAN: a Self-Adapted System for Anomaly Detection in Smart Buildings Data Streams

Maxime Houssin, Stéphanie Combettes, Marie-Pierre Gleizes, Bérangère
Lartigue

► **To cite this version:**

Maxime Houssin, Stéphanie Combettes, Marie-Pierre Gleizes, Bérangère Lartigue. SANDMAN: a Self-Adapted System for Anomaly Detection in Smart Buildings Data Streams. 18th Adaptive Computing (and Agents) for Enhanced Collaboration at IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (ACEC @ WETICE 2020), Jun 2020, Bayonne, France. hal-03024015

HAL Id: hal-03024015

<https://hal.science/hal-03024015>

Submitted on 25 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SANDMAN: a Self-Adapted System for Anomaly Detection in Smart Buildings Data Streams

Maxime Houssin
IRIT & LMDC

University of Toulouse III
- Paul Sabatier, France
maxime.houssin@irit.fr

Stéphanie Combettes
IRIT

University of Toulouse III
- Paul Sabatier, France
stephanie.combettes@irit.fr

Marie-Pierre Gleizes
IRIT

University of Toulouse III
- Paul Sabatier, France
marie-pierre.gleizes@irit.fr

Berangere Lartigue
LMDC

University of Toulouse III
- Paul Sabatier, France
berangere.lartigue@univ-tlse3.fr

Abstract—Currently, energy management within buildings is essential to mitigate climate change. To this end, buildings are increasingly equipped with sensors to assist the building manager. Yet, the heterogeneity and the large amount of generated data make this task quite difficult. The SANDMAN multi-agent system, described in this paper, aims to assist in the automatic detection, in constrained time, of several types of anomalies using raw and heterogeneous data. SANDMAN features a semi-supervised learning by considering some feedback from an expert in the field. The results show that SANDMAN detects different types of anomalies, is resilient to noise and is scalable.

Index Terms—Self-Adaptive Multi-Agent System, Anomaly Detection, Smart Buildings

I. INTRODUCTION

Buildings account for more 30% of worldwide final energy consumption, this figure rising to over 40% in developed countries. However, much of this energy is wasted - 30 % in the United States and Europe. Among other reasons, this waste comes from poor building management, such as the failure to quickly detect problems.

The objective of the work presented in this article is to detect anomalies in energy data issued from smart buildings to quickly solve the problems and minimise the energy waste.

With the advent of IoT (Internet of Things), the number of sensors in existing and new buildings have sharply risen due to their lower costs and the obvious benefit of using them for building management. Sensors can easily be added to buildings or replaced by others. Therefore, managing these sensors and the data they generate is a complex system for building managers.

However, an accurate control of these data is necessary for the proper management of buildings, including their energy performance. This is why a tool for automatic detection of anomalies is an important asset for building managers. This tool must process the data in constrained time to allow to act as soon as possible, which is an essential issue in the detection of anomalies. Because of the large amount of data, the spaces for searching for anomalies also become very large: this is the reason why the system must learn to raise anomalies in a semi-supervised manner thanks to the feedback of an expert of the field. The system detects and gives the found anomalies to the expert and the expert gives feedback to allow the system to learn throughout its life.

The article is composed as follows. Section I presents the characteristics required for anomaly detection systems and defines the notions of anomalies. Section II describes the main existing anomalies detection systems. Section III explains the architecture of the proposed multi-agent system, SANDMAN (Semi-Supervised ANomaly Detection with Multi-AgeNt systems). Section IV is devoted to SANDMAN's experiments and results. Finally, Section V concludes and proposes perspectives to the presented work.

A. Required Characteristics for an Anomaly Detection System

An intelligent building equipped with sensors generates a large amount of available data that must be analysed to improve energy management. For example, the SGE (*Service de Gestion et d'Exploitation* in charge of energy management for the campus of University of Toulouse III-P. Sabatier) manages about 6000 sensors measuring at least one value per hour. Thus, the design of a smart building anomaly detection system must ideally take into account the following characteristics:

- interaction with an expert,
- constrained time detection,
- detection of several types of anomalies,
- use of raw data,
- use of heterogeneous data,
- openness,
- scaling,
- genericity.

B. Categorisation of the Anomalies

Each sensor provides timestamped data. For each one, the **actual value** is the measured data. A **nominal value** is calculated from a profile based on the values usually provided by the sensor. The location of the sensors is unknown, as well as any other metadata (type of sensor, acquisition frequency, etc.). As the lack of information on the sensors is a common problem [1], it is necessary to take them all into account in the method of anomaly detection.

Chandola et al. [2] define an anomaly as an unexpected or undesirable behaviour in a system. The authors define 3 kinds of anomalies:

- **point**: a measure is outside of an acceptable range for the sensor,

- **contextual**: a measure is inside an acceptable range for the sensor but anomalous in some context (example: high heating consumption during summer),
- **collective**: a collection of measures is anomalous with respect to the entire data set although the individual measures may not be anomalous in themselves.

row	timestamp	sensor1		sensor2		sensor3		sensor4		sensor5		anomaly
		measured	nominal	measured	nominal	measured	nominal	measured	nominal	measured	nominal	
0	2016-01-11T00:00	16	17	11	10	160	160	11	11	66	66	
1	2016-01-11T01:00	15	15	11	11	140	140	-5	11	64	65A	
2	2016-01-11T02:00	15	15	12	12	130	127	10	10	65	65	
3	2016-01-11T03:00	15	16	12	13	110	110	8	8	64	65	
4	2016-01-11T04:00	15	14	12	12	120	120	9	9	65	65	
5	2016-01-11T05:00	15	15	14	14	140	140	13	13	400	65	
6	2016-01-11T06:00	16	17	16	14	170	165	17	17	66	66	
7	2016-01-11T07:00	17	19	20	20	220	200	25	25	67	67	
8	2016-01-11T08:00	23	19	19	23	220	235	34	29	68	67A	
9	2016-01-11T09:00	18	18	25	25	260	255	33	33	68	68	
10	2016-01-11T10:00	19	18	27	27	230	245	35	34	69	69	
11	2016-01-11T11:00	19	19	25	25	210	235	32	32	69	70	
12	2016-01-11T12:00	21	21	26	24	210	230	32	32	71	71	
13	2016-01-11T13:00	22	22	28	28	220	240	34	34	72	72A	
14	2016-01-11T14:00	22	22	29	29	270	270	34	33	72	72	
15	2016-01-11T15:00	23	23	31	31	270	270	35	35	73	73	
16	2016-01-11T16:00	24	24	32	29	270	270	35	35	76	74	
17	2016-01-11T17:00	23	23	29	29	250	258	31	31	73	73	
18	2016-01-11T18:00	22	24	29	29	240	240	31	30	72	72	
19	2016-01-11T19:00	21	22	26	26	220	220	27	27	71	71	
20	2016-01-11T20:00	19	19	22	22	200	200	23	23	69	69	
21	2016-01-11T21:00	19	20	21	22	190	185	21	19	70	69	
22	2016-01-11T22:00	18	18	20	20	180	180	20	20	68	68	
23	2016-01-11T23:00	17	17	16	16	170	170	16	16	67	67	

TABLE I
EXAMPLE OF DATA AND EXPERT'S FEEDBACK

Table I illustrates the 3 types of anomalies defined above. They are indicated by the letter A in the last column. Row 1 is considered anomalous by the expert. Sensor 4 has a value of -5, which is far from the nominal value of 11. This is a point anomaly.

Row 8 is an anomaly because of the values of sensors 1 to 4. Although taken individually, none of these values would be considered an anomaly, together they generate a collective anomaly.

The last anomaly in row 13 appears at 13:00 and only then. This is due to the fact that the actual values of sensor 3 are disparate from line 9 to line 13. In real life, an expert may have noticed a problem at 13:00 due to the constant accumulation of small differences in the previous 4 lines. This is called a contextual anomaly because it is necessary to take into account the recent past to detect it.

Row 5 is not an anomaly even if the value of 400 in sensor 5 is very far from the nominal value of 65. It is not considered as an anomaly by the expert, because this sensor is either irrelevant for the detection of anomalies or that the variation is not large enough for an expert to consider it as an anomaly.

II. STATE OF THE ART OF ANOMALY DETECTION IN SMART-BUILDINGS

Given the real challenge of detecting anomalies in buildings, the field of research in this area is rapidly expanding. Multi-agent system (MAS) has been introduced in building physics over the last ten years or so, but their applications are almost entirely dedicated to optimising system management (heating, cooling, ventilation) or the entire building [3]. No study, to our knowledge, has been conducted for an application in the detection of anomalies in building data, even though MASs have been used to detect anomalies in other domains [4] [5].

Methods used in the literature include **physical models**, which require modeling the system or building and comparing

it with measured data. These methods are quickly limited, given the complexity of the systems to be modeled [6].

Unsupervised and statistical classification methods have been used to detect anomalies in building data [7] [8] [9]. However, these studies are only based on a small amount of data and do not show that scaling is possible. They are also based on the assumption that the number of anomalies is small relative to the total number of data.

Neural networks are tools widely used in building studies, but rarely for the detection of anomalies [10] [11]. Although neural networks can detect anomalies in real time, a large set of labeled data is required to train the networks. However, this step seems impossible to perform for the size of systems considered in this study and it does not correspond to what actually happens in building management.

The methods of **data mining** are used to detect anomalies in building data [12] [13] [14]. However, a data pre-processing step is required, as well as an expert for the selection of relevant data during scaling.

III. SANDMAN ARCHITECTURE

A. Definitions of Situation and Sensor Profile

SANDMAN works with all the available sensor types and thanks to the feedback given by a human expert. For example, in Table I, SANDMAN has access to the values of the 5 sensors. We define the notion of **situation** as the set of values measured by all these sensors over 24 hours and we call **current situation** the last situation encountered. The last measurement of all the sensors is called **instantaneous values of the sensors**: this corresponds to the last row in Table I, containing the instantaneous values of the 5 sensors at 23:00. The situations encountered by SANDMAN are stored in a situation history.

Each sensor is associated with a **profile**, consisting of 24 measured values, one for each hour of the day. This profile indicates the value expected at any time of the day for the sensor in question. It is updated by SANDMAN throughout the life of the system.

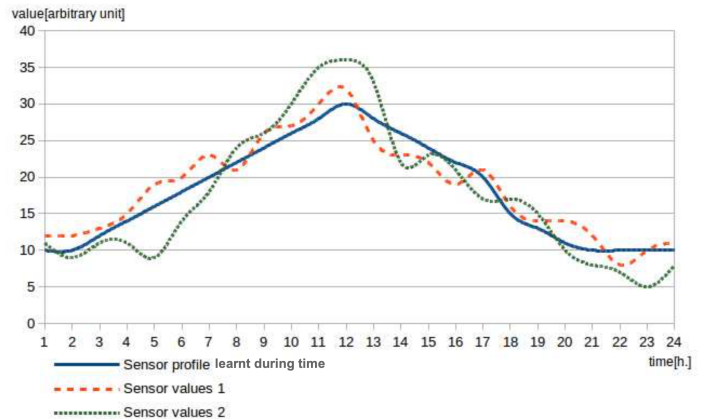


Fig. 1. 2 sets of measured values from the same sensors over 24h

Figure 1 shows the profile of one sensor as a function of time over a period of 24 hours. The measured values *values 1* are a possible example of measured values for this sensor and the measured values *values 2* are another possible example of measured values for this sensor. The measured values *values 1* and the measured values *values 2* are examples and of course cannot be measured at the same time for the same sensor. It can be seen that the measured values *values 1* are close to the expected 24-hour profile while the measured values *values 2* are considered anomalous.

B. Detection of Anomalies by SANDMAN

Since all available sensors are used and the expert may only be interested in certain faults, and since each sensor has its own interval and value range, profile learning alone is not sufficient to detect anomalies. Indeed, some sensors may not be relevant for the detection of anomalies. This is the case, for example, of sensors located in a building that the expert is not interested in, or if the expert is only interested in power consumption then an air quality sensor will not be relevant. Lastly, for sensors that are actually useful, depending on the sensors, some deviations from expected values are greater than others. This is due to the fact that their values are not normalised because SANDMAN uses raw data.

To raise or not to raise an anomaly at each new current situation, SANDMAN uses both the measured values of the sensors over 24 hours and the profiles of the associated sensors. To do this, SANDMAN calculates **the disparity of a sensor** which is the sum of all the differences between the sensor profile and the sensor measured values over the last 24 hours, i.e. for the current situation (Eq.(1)).

$$Disparity_s^t = \sum_{t_i=t}^{t-23} |realValue_s^{t_i} - nominalValue_s^{t_i}| \quad (1)$$

with :

- s : sensor s ;
- t : time t of the current situation;
- $realValue_s^{t_i}$: real value of sensor s at time t_i ;
- $nominalValue_s^{t_i}$: nominal value of sensor s at time t_i ;

The period under consideration is a sliding 24-hour window. If the time of the current situation is 15:00, all data from 16:00 of the previous day to 15:00 of the current day are considered for the profile. SANDMAN then calculates **the Degree of Anomaly (DA) of the current situation** which is the weighted sum of the measured values of all sensors (Eq. (2)).

$$DA(Situation^t) = \sum_{s=1}^S Disparity_s^t * Weight_s \quad (2)$$

with :

- $Situation^t$: Situation at time t ;
- S : number of sensors;
- $Disparity_s^t$: Disparity of the sensor s at time t ;

- $Weight_s$: Weight of the sensor s .

Thus thanks to the degree of anomaly, SANDMAN classifies the current situation, i.e. labels it "normal" or "anomalous" by algorithm 1. The threshold used is a static number

Algorithm 1 Algorithm of classification or anomalies detection

```

1: for each new current situation do
2:   if DA < Threshold then
3:     SANDMAN returns : "the situation is normal"
4:   else
5:     SANDMAN returns : "the situation is anomalous"
6:   end if
7: end for

```

chosen arbitrarily (1000 for SANDMAN). The calculation of each weight adjusts to the threshold whatever its value, and SANDMAN classifies the situation (on the correct side of the threshold) according to the value of this weight. Therefore, the DA also depends of the threshold value.

To correctly detect anomalies, the $Weights_s$ associated with the sensors must therefore have the correct values. A sensor has only one associated weight that is used to calculate the degree of anomaly for all situations in the situation history.

C. Learning by SANDMAN

Once SANDMAN has classified a situation, i.e. labeled it as "normal" or "anomalous", and received feedback from the expert, it performs a resolution cycle in which it analyses whether or not it should learn. SANDMAN therefore analyses the following cases:

- 1) For a situation classified as "anomalous" by SANDMAN and "anomalous" by the expert,
 - a) SANDMAN does nothing
- 2) For a situation classified as "normal" by SANDMAN and "anomalous" by the expert,
 - a) SANDMAN creates and adds a new "anomalous" situation to the situation history
 - b) SANDMAN self-adapts the weights
- 3) For a situation classified as "anomalous" by SANDMAN and "normal" by the expert,
 - a) SANDMAN updates the sensor profiles
 - b) SANDMAN creates and adds a new "normal" situation to the situation history
 - c) SANDMAN self-adapts the weights
- 4) For a situation classified as "normal" by SANDMAN and "normal" by the expert,
 - a) SANDMAN updates the sensor profiles.

The main steps for updating profiles, adding a situation in the history and self-adjusting weights are described below.

Profile updates. They are only updated if the situation is labelled "normal" by the expert (steps 3a and 4a), because the sensor values in anomalous situations are not reliable and do

not represent the expected values. The update formula applied to each sensor profile is as follows (Eq.(3)) :

$$val_{profile}^{tt} = (1 - \lambda) * val_{profile}^t + \lambda * val_{sensor}^t \quad (3)$$

with

- t : time of the current situation,
- $\lambda \in]0,1[$ represents the importance of the new value versus the previously learned profile value,
- $val_{profile}^t$: the value previously learned and stored in the profile at time t
- val_{sensor}^t : the real value measured at time t
- $val_{profile}^{tt}$: the new value learned and stored in the profile at time t .

Each sensor profile modifies one of its 24 values corresponding to the time t of the current situation using the measured value val_{sensor}^t of the current situation.

Adding a situation to the situations history. This operation, performed in steps 2a and 3b, initiates SANDMAN resolution cycles that occur until all situations in the history are correctly and evenly classified. A situation is classified correctly if its degree of anomaly (DA) enables to deduce the correct classification (normal or anomalous) of the situation. Situations have to be classified in a balanced manner: thus the degree of anomaly of the "normal" situation closest to the threshold and the degree of anomaly of the "abnormal" situation also closest to the threshold are, in absolute value, at the same distance from the given threshold. This is illustrated in Figure 2, where the points respectively above and below the threshold $Threshold$ and closest to the threshold are at equal distance from the threshold.

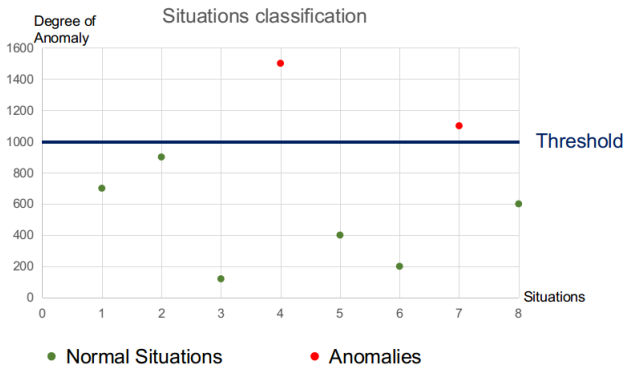


Fig. 2. Example of 8 classified situations

Each situation is assigned a criticality value according to its degree of anomaly and its classification given by the expert. Criticality represents the degree of satisfaction of the situation with respect to its objective. A situation labeled "normal" aims at a degree of anomaly as low as possible with respect to the $Threshold$ threshold, while a situation labeled "anomalous" aims at a high degree of anomaly with respect to the $Threshold$ threshold. The criticality $Crit$ is calculated

as for a "normal" situation (Eq.(4)) and for an "anomalous" situation (Eq.(5)).

$$Crit_{Situation^t} = DA(Situation^t) - Threshold \quad (4)$$

$$Crit_{Situation^t} = Threshold - DA(Situation^t) \quad (5)$$

Every situation wants to minimise its criticality, a negative criticality meaning that the situation is correctly classified. Since disparities and weights are always positive, situations labeled "normal" and situations labeled "anomalous" have antagonistic objectives, wanting respectively higher or lower weights. When a resolution cycle is performed, each situation sends a message to each weight agent with:

- the desired direction of weight change (lower for "normal" situations, higher for "anomalous" situations);
- the criticality of the situation;
- the influence of the weight in the degree of anomaly.

The influence $Influ$ of a sensor s in a situation t corresponds to the importance of the sensor weight on the sum of the weight of all the sensors for this situation, as shown in eq.(6):

$$Influ_s^t = \frac{Weight_s * Disparity_s^t}{DA(Situation^t)} \quad (6)$$

A resolution cycle starts when situations require the weights to adjust. The cycle ends when the weights have self-adjusted.

Self-adapting weights. Each weight is represented by a weight agent. The goal of the multi-agent weight system is that each weight agent finds its value by cooperating with the others. During a resolution cycle, all situations require weight adjustments (steps 2b and 3c) and the weight agents self-adapt as described in the algorithm 2, then each situation calculates its new degree of anomaly using the updated weights.

On receiving the parameters sent by the situations (direction, criticality and influence), the weight agents decide, or not, to update independently and simultaneously their weights. To do so, they follow the algorithm 2.

Algorithm 2 Self-adaptation of the *Weight* agent of the sensor s

Require: *Direction, Crit, Influ* of all situations

- 1: Select the most critical "normal" situation NS
 - 2: Select the most critical "anomalous" situation AS
 - 3: **if** $Crit(NS) > Crit(AS)$ **then**
 - 4: **if** $Influ_s^{NS} > Influ_s^{AS}$ **then**
 - 5: The *Weight* agent reduces its weight
 - 6: **end if**
 - 7: **if** $Influ_s^{AS} > Influ_s^{NS}$ **then**
 - 8: The *Weight* agent raises its weight
 - 9: **end if**
 - 10: **end if**
-

The cooperation between the weight agents guarantees that at least one weight agent updates its weight and that each update cycle of the weight agents results in a better general state where the most critical weight agent decreases its criticality [15]. After each updating cycle of the weight

agents, the situations calculate their new criticality by taking into account the new values of the weight agents. This process is repeated until the most critical situations labeled "normal" and "anomalous" have their respective criticalities both equal (within a small margin of error) and negative. This ensures that all situations are correctly classified and that the following situations have a better chance of being classified accordingly by SANDMAN. Indeed, the best way to reduce the rate of false negatives and false positives [2], and thus to have a robust system, is to keep the most critical known situations (saved in the history) as far from the threshold as possible.

IV. EXPERIMENTATIONS

A. Description of the Experimental Framework

The data used to conduct the experiments were generated using the TSimulus time series generator [16]. These time series include a signed value for each sensor for each hour. Each sensor has its own range of values. The values for each sensor are cyclical, with or without noise over a 24-hour period, i.e. the value of a sensor (without noise) at 3 p.m. is the same every day. These simulated data are modified by a human expert to introduce anomalies of different types. SANDMAN processes the data in constrained time and the expert gives his opinion a posteriori in an asynchronous way.

All the experiments were performed on a 4-core processor with a 2.6 GHz frequency. The data sets used in the following experiments feature one value for each sensor every hour over one month, for a total of 744 hours. The number of sensors depends on the experiment. Four separate data files are used to show the ability of SANDMAN to 1) detect point anomalies and mitigate noise, 2) detect collective and contextual anomalies, 3) adapt the profiles of sensors, 4) scale.

In the experiments, we chose to duplicate the one-month data set in order to show the improvement granted by the learning of SANDMAN. Thus, the results are presented after one month and two months, as the number of:

- TP: True Positive, TN: True Negative,
- FN: False Negative, FP: False Positive,
- t/sit : calculation time for processed situation in ms.

B. Point Anomalies and Noise Mitigation

Point anomalies are the type of anomaly most often detected in the literature, as both univariate and multivariate approaches can detect them.

In this experiment, we study the efficiency of SANDMAN in detecting point anomalies in noisy data with different levels of noise. The data file contains 20 sensors, i.e. $20 \times 744 = 14880$ data. 58 point anomalies were added manually by an expert. Three experiments have been conducted, with the noiseless data, 1%-level data and 5%-level data.

The noise is simulated by applying a uniform distribution to each sensor with the following formula (Eq (7)):

$$Value_{noise} = Value_{noNoise} - \frac{|MIN(s) - MAX(s)|}{2} + rand_{[0,1]} * interval * noise \quad (7)$$

Noise	Month 1				Month 2			
	TP	TN	FN	FP	TP	TN	FN	FP
No noise	40	686	18	0	58	686	0	0
1%	40	686	18	0	58	686	0	0
5%	58	655	0	31	58	679	0	7

TABLE II
RESULTS OF POINT ANOMALIES DETECTION WITH NOISY DATA

	nb	Month 1		Month 2	
		TP	FN	TP	FN
Point	6	2	4	6	0
Collective	5	1	4	5	0
Contextual	3	1	2	3	0
Total	14	4	10	14	0

TABLE III
RESULTS OF ANOMALY DETECTION PER KIND OF ANOMALY

with *noise* : quantity of noise, respectively 0%, 1% and 5% in the experiments.

Table II shows the results of the 3 experiments after month 1 and month 2. We can see that SANDMAN improves its detection with time. After month 2, it detects the right anomalies (58 TP) when the data have a low noise level. However, a higher noise level leads to a higher false positive rate (7 FP after month 2 in the case of 5% noise). SANDMAN reduced this false positive rate along time by creating normal situations, and thus by learning lower weights, as shown in Algorithm 2.

C. Collective and Contextual Anomalies

In this experiment, we want to study the ability of SANDMAN in detecting different types of anomalies, such as point, contextual and collective anomalies. Contextual anomalies occur when a sensor value is unusual several times in a row. Collective anomalies occur when several sensor values are unusual at the same time, but not so unusual that only one of them causes a single point anomaly. In this experiment, the data file contains the values of 20 sensors over one month, and the results are presented after months 1 and 2. The 3 types of anomalies (point, collective and contextual) have been manually added in the data file, for a total of 14 anomalies. The data have 1% of noise.

Table III shows the results of this experiment. The numbers of each type of anomaly, as well as the anomalies detected after the month 1 and then after the month 2. After month 2, SANDMAN detected the right anomalies. There are no false positives in the results of this experiment. SANDMAN created and added to the history the situations corresponding to the misclassified situations during the month 1 to classify them without error during the month 2, regardless of the type of anomaly.

D. Updating Profile on Sliding Data

In previous experiments, the profile of each sensor was stored and contributed to the detection of anomalies, but the

	Month 1				Month 2			
	TP	TN	FN	FP	TP	TN	FN	FP
Sliding values	46	686	12	0	58	686	0	0

TABLE IV
RESULTS OF PROFILE ADAPTATION

Number of sensors	Month 1 t/sit (ms)	Month 2 t/sit (ms)
20	2.7	1.7
40	4.8	3.5
100	22	20
200	48	45
400	102	100
800	185	180

TABLE V
RESULTS OF SCALING

data was cyclical on a daily basis. In this experiment, the measured values of the sensors are modified so that each measured value is 1% higher each day, so that over a 31-day month, each sensor has measured values that are 31% higher than the control data. The test file contains 58 point anomalies and each of the 20 sensors causes an anomaly at least once. Table IV presents the results of the experiment. The update of the sensor profile is sufficiently reactive to allow SANDMAN to classify anomalies when the measured values of the sensors change over time.

E. Scaling

In this experiment, an increasing number of sensors is used to measure this effect on the computing time. To do this, a data set of 20 separate sensors is duplicated to obtain up to 800 sensors. The data used is the same as for the previous experiment on rolling data without the change in values of 1% per day. Table V shows the calculation time per situation as a function of the number of sensors. The calculation time of month 2 is always shorter than the calculation time of month 1 because the learning of the weights is only carried out in month 1. We also note that the resolution time is proportional to the number of sensors used and that the time difference between the two months is constant. This is due to the fact that most of the execution time comes from reading the raw data from a database, which has a fixed cost per sensor.

V. CONCLUSION AND PERSPECTIVES

For the management of energy-related data in an intelligent building, several types of anomalies must be detected. They can be point, collective, or contextual anomalies. The detection of these different types of anomalies is absent from the state of the art methods for which data pre-processing is mandatory.

We, therefore, proposed SANDMAN, a semi-supervised real-time anomaly detection system that uses raw data as input and classifies anomalies by learning from the expert's feedback. SANDMAN is able to detect three kinds of anomalies in a generic way and is well suited to a growing number of sensors. The feedback from the expert is reduced to a

minimum because it is optional and SANDMAN can be operated without prior labelling.

In future works, after an anomaly occurs the system must be able to observe the return to nominal values of the sensors to inhibit instantaneously the anomalies. This is the improvement we are currently developing. SANDMAN will also have to be able to learn several sets of profiles for the sensors, in order to take into account the different behaviours of building users (for example, weekdays and weekend periods).

REFERENCES

- [1] J. I. Guerrero, A. García, E. Personal, J. Luque, and C. León, "Heterogeneous data source integration for smart grid ecosystems based on metadata mining," *Expert Systems with Applications*, vol. 79, pp. 254–268, aug 2017.
- [2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," vol. 41, no. 3, pp. 1–58, 2009.
- [3] T. Labeodan, K. Aduda, G. Boxem, and W. Zeiler, "On the application of multi-agent systems in buildings for improved building operations, performance and smart grid interaction – A survey," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 1405–1414, 2015.
- [4] A. Forestiero, "Self-organizing anomaly detection in data streams," *Information Sciences*, vol. 373, pp. 321–336, 2016.
- [5] Y. Seng Ng and R. Srinivasan, "Multi-agent based collaborative fault detection and identification in chemical processes," *Engineering Applications of Artificial Intelligence*, vol. 23, no. 6, pp. 934–949, 2010.
- [6] W. J. Turner, A. Staino, and B. Basu, "Residential HVAC fault detection using a system identification approach," *Energy and Buildings*, vol. 151, pp. 1–17, 2017.
- [7] C. Miller, Z. Nagy, and A. Schlueter, "A review of unsupervised statistical learning and visual analytics techniques applied to performance analysis of non-residential buildings," jan 2018. doi: 10.1016/j.rser.2017.05.124.
- [8] K. Yan, Z. Ji, and W. Shen, "Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM," *Neurocomputing*, vol. 228, no. July 2016, pp. 205–212, 2017.
- [9] J.-S. Chou and A. S. Telaga, "Real-time detection of anomalous power consumption," *Renewable and Sustainable Energy Reviews*, vol. 33, pp. 400–411, may 2014.
- [10] J. Wu, W. Zeng, and F. Yan, "Hierarchical Temporal Memory method for time-series-based anomaly detection," *Neurocomputing*, vol. 273, pp. 535–546, 2018.
- [11] Y. Zhu, X. Jin, and Z. Du, "Fault diagnosis for sensors in air handling unit based on neural network pre-processed by wavelet and fractal," *Energy and Buildings*, vol. 44, pp. 7–16, jan 2012.
- [12] A. Capozzoli, F. Lauro, and I. Khan, "Fault detection analysis using data mining techniques for a cluster of smart office buildings," *Expert Systems with Applications*, vol. 42, pp. 4324–4338, jun 2015.
- [13] M. Peña, F. Biscarri, J. I. Guerrero, I. Monedero, and C. León, "Rule-based system to detect energy efficiency anomalies in smart buildings, a data mining approach," *Expert Systems with Applications*, vol. 56, pp. 242–255, sep 2016.
- [14] P. Xue, Z. Zhou, X. Fang, X. Chen, L. Liu, Y. Liu, and J. Liu, "Fault detection and operation optimization in district heating substations based on data mining techniques," *Applied Energy*, vol. 205, pp. 926–940, nov 2017.
- [15] J.-P. Georgé, M.-P. Gleizes, and V. Camps, "Cooperation," in *Self-organising Software* (G. Di Marzo Serugendo, M.-P. Gleizes, and A. Karageorgos, eds.), Natural Computing Series, pp. 193–226, <http://www.springerlink.com>: Springer, 2011.
- [16] "Tsimulus." <https://tsimulus.readthedocs.io/en/latest/>. Accessed: 2019-12-18.