A Machine Learning Approach for Identifying and Classifying Faults in Wireless Sensor Networks

Warriach, Ehsan Ullah; Aiello, Marco; Tei, Kenji

Published in:
International Conference on Computational Science and Engineering

2012 IEEE 15th International Conference on Computational Science and Engineering

A Machine Learning Approach for Identifying and Classifying Faults in Wireless Sensor Networks

Ehsan Ullah Warriach, Marco Aiello
Department of Mathematics and Computer Science
University of Groningen
Groningen, The Netherlands
Email: e.u.warriach,m.aiello@rug.nl

Kenji Tei
National Institute of Informatics
Tokyo, Japan
Email: tei@nii.ac.jp

***Abstract*—Wireless Sensor Network (WSN) deployment experiences show that collected data is prone to be faulty. Faults are due to internal and external influences, such as calibration, low battery, environmental interference and sensor aging. However, only few solutions exist to deal with faulty sensory data in WSN. We develop a statistical approach to detect and identify faults in a WSN. In particular, we focus on the identification and classification of data and system fault types as it is essential to perform accurate recovery actions. Our method uses Hidden Markov Models (HMMs) to capture the fault-free dynamics of an environment and dynamics of faulty data. It then performs a structural analysis of these HMMs to determine the type of data and system faults affecting sensor measurements. The approach is validated using real data obtained from over one month of samples from motes deployed in an actual living lab.**

I. Introduction

Wireless Sensor Networks (WSNs) has been extensively employed for enabling various monitoring and control applications such as environment surveillance, industrial sensing, or traffic monitoring [6]. Numerous mobile and pervasive applications are constantly collecting and processing information from the physical world and providing information about sensed environment or events at a high level of detail. Long-term deployments of WSNs in real world settings are becoming more frequent, also because wireless sensor network software and hardware are progressing dramatically [16]. The cornerstone for the success lies in the ability to draw meaningful and precise inferences from the collected data, which in turn requires to have high sensor data quality.

WSNs are installed for the purpose of sensing and monitoring an area of interest for specific physical quantities, often implying that these wireless sensors are left unattended for long periods of time in the field, and in turn rendering them prone to failures. Cheap sensors are also incline to develop faults as they age posing a major problem for the application, as the data from the network becomes progressively unreliable. An early detection of such fault is necessary for the effective operation of the sensor network. Many deployment experiences show that data collected from WSNs are prone to be faulty due to internal and external influences, such as calibration, battery drain, environmental interference and sensor aging.

As wireless sensor network technology progresses, ensuring data quality needs also to become an active area of investigation. For instance, with the aim of creating a simple to use wireless sensor network application, in [22], the authors perceived the complexity of collecting correct sensor data. A key source of faults in WSNs is calibration. Sensor nodes during installed periods possibly will offset or gain, and it is a significant effort to make them correct. The authors of [22] conclude that calibration is a hard challenge for the upcoming progress in WSN. They examine that faults can happen in surprising ways. For example, Tolle et al. [13] installed a WSN application by using the system explained in [22], to investigate the microclimate over the volume of a redwood tree. The authors found that only 49% of the data samples will possibly be considered as significant because there are numerous faulty data sample readings that need to be disregarded. Two recent papers [23], [2] proposed approaches to execute calibration on-line, though there are some concerns as the deployed wireless sensor network lacks any ground truth samples for comparison. Generally, wireless sensor nodes experience two broad categories of faults; those affecting the performance of WSN such as, *system and data faults*. On the one hand, there is the *data centric view* comprising faults such as *stuck-at, offset* and *gain*. On the other hand, there is the *system centric view* with faults such as *calibration, low battery* and *environment out of range*.

Surprisingly, the requirement for consistent and protected data collection in sensor applications appears to have received less attention than the problem of protecting a WSN application from network-level system faults, e.g., malicious message routing. A limited number of studies have examined to manage WSN from the effects of faulty sensor data, an exception being [7]. This paper presents an on-the-fly statistical approach to distinguish faulty sensor readings and to identify and classify data and system fault types in a WSN. Identifying data and system faults is necessary to confirm the precise accuracy of a WSN, while differentiating data faults from system faults is essential to identify the causes of failure and to pledge a precise recovery action. The proposed method considers Hidden Markov Models (HMMs) to detect and identify data and system faults types. At every time step of the algorithm we propose, multiple correlated observations are gathered from several sources. Assuming that the faults have not conceded

978-0-7695-4914-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICCSE.2012.90

yet into the sources, we use a statistical clustering-based method to statistically distinguish accurate observations from faulty ones. Therefore, we can efficiently identify the HMM capturing the correspondence between the hidden and accurate changes of the detected phenomenon (based on data from fault-free sensors) and the observable changes of the sensed phenomenon (based on fault and fault-free sensory data).

Previous work has used HMMs to simply detect variances, here we make a broader use of it by considering the identification and classification of the type of the detected data fault such as *stuck-at*, *offset* and *gain*, and also to identify and classify the system faults that affect the sensor network. To achieve this goal, we need another HMM that represents the coherence among the hidden changes of detected phenomenon and the changes of data and system faults, and through a structural reasoning between the two HMMs. We validate the approach using real world data samples collected over 15 days of readings from motes deployed in lab created in connection to an European Framework seven project called GreenerBuildings [4].

The rest of the paper is organized as follows. The basics of HMMs are recalled in Section II. The proposed approach for detecting and classifying data and system faults in WSNs is presented in Section III. In Section IV, we define common data and system fault models and their causes at the network and node levels. The data and system faults classification method is presented in Section V. The experimental results of the proposed method with real-world dataset is presented in Sections VI. Section VII discusses related work. Finally, in Section VIII we provide our concluding remarks.

II. Hidden Markov Models

A HMM captures a stochastic process that is concluded through a sequence of observations, which are stochastically related to the state of the hidden process [3]. A HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. In an HMM, the state is not directly observable, but variables influenced by the state are observable. Each state has a probability distribution over the possible output observations. Therefore the sequence of observations generated by an HMM gives some information about the sequence of states. Mathematically, an HMM is characterized by:

- N: The number of states in the model. We represent the single states as $S = \{s_1, s_2, ...s_N\}$.
- M: The set of possible measurements. We represent the single measurement as $V = \{v_1, v_2, ..., v_N\}$.
- C: A sample symbol probability distribution $C = \{c_{gi}\}$, where $c_{gi} = P\{v_t = V_i | s_t = S_g\}$ and v_t denote the sample reading at time t, $1 \leq g, i \leq N$.
- D: The state transition probability $D = \{d_{gh}\}$, where d_{gh} characterizes the probability of a transition to state h from state g. $d_{gh} = P\{s_{t+1} = S_h | s_t = S_g\}$ and s_t is the hidden state at time t, $1 \leq g, h \leq N$.
- π: The initial state distribution $\pi = \{\pi_g\}$, where π_a is the probability that the HMM starts in state a: $\pi_g = P\{s_v = S_g\}$, $1 \leq g \leq N$.

In order to estimate the parameters N, M, C, D and π of the HMM, we use a supervised learning technique (Figure 1). Where a dataset is partitioned into training and test sets. We inject data faults into (fault-free) training dataset, label each sample as fault-free or faulty with a particular fault type, and use this labeled data for estimating the parameters of the HMM. In an early fault-free training period, a HMM

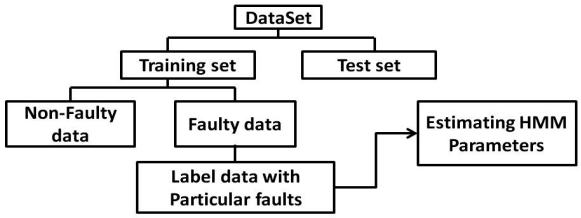

Fig. 1: Environment Modeling through HMMs

σ is recognized to represent the accurate behavior of the application.

III. Identifying and Classifying Faults

Data samples from sensors deployed in an area of interest are used as the input to the client application on the base station for analysis, then on a single base station the following steps are taken (see Figure 2):

- Accumulate sensory data and combine it based on a predefined time window T_w.
- From the samples collected in each time window and a set of possible states of the environment create: (i) a set v_g of the visible states of the sensed phenomenon (based on the data irrespective of their precision), (ii) a set a_g of the hidden states of the sensed phenomenon, and (iii) a set f_g of the faulty states obtained from samples those degraded as a result of system faults.
- Construct a HMM_{AV} that describes the hidden (denoted with a_g) and visible states of the environment (denoted with v_g), and a HMM_{AF} that describes the hidden (denoted by a_g) with the fault states (denoted by f_g).
- Examine the two HMMs, which are based on known fault models, identify the type of data and system fault that has affected the sensor samples.
- Construct a Markov Model M_A showing a fault-free states of the sensed environment to the user.

A. Data Collection

Data streams from sensors are used as the input to the client application at the base station for analysis, as shown in Figure 2. In general, sensors are multimodal and measure several physical quantities. For example, we use the CM5000 mote, which is an IEEE 802.15.4 compliant wireless sensor node based on the original open-source "TelosB" platform. The included sensors measure temperature, relative humidity, and

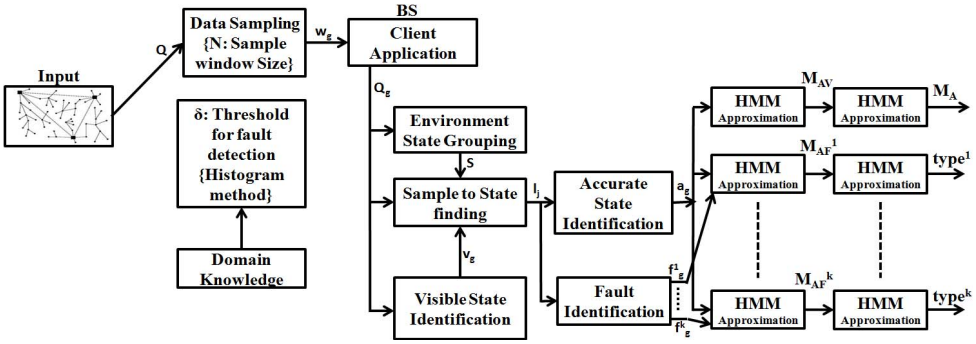

Fig. 2: Data and System Fault Detection Approach

light. The value of the target environment attributes examined with sensors is denoted by $Q(t)$ in the deployed space of interest as a multidimensional, unidentified constraint that differs by time.

We assume that each sensor periodically sends a message (t,x) to a base station, where t is the time of reading of the value x from a sensor g such that $x_g = Q(t) + \varepsilon_g$, with ε_g denotes the additive noise. Suppose that a sensor sample $V = \{(t,x)\}$ is measured at a base station and split the given sample into intervals of period T_w to have an array of measured samples $\{V_g\}$ such that:

$$V_g = \{x|(t,x) \in V \wedge T_w.(g-1) \leq t \geq T_w.g\} \quad (1)$$

The constraint T_w have to be large enough so that V_g are non-empty sets. An *environment state grouping* component uses an online statistical clustering algorithm at given samples to recognize the likely states $S = \{s_1, s_2 ..., s_N\}$ of the environment. Figure 3 shows an example where six states s_h are recognized. A *visible state identification* component concludes the current visible state of the given sample v_g by using a present samples set V_g. The visible state defines the entire samples readings $x_1, x_2 ..., x_N$ in V_g:

$$v_g = \arg \min_{1 \leq i \leq M} ||s_i - \frac{1}{N} \sum_{h=1}^{N} x_h|| \quad (2)$$

Figure 3 shows an example where a sample set V_g of five readings x_h is mapped onto the visible state s_3, since s_3 is the nearby state with respect to the mean value measured through given readings. A *sample to state finding* component finds all

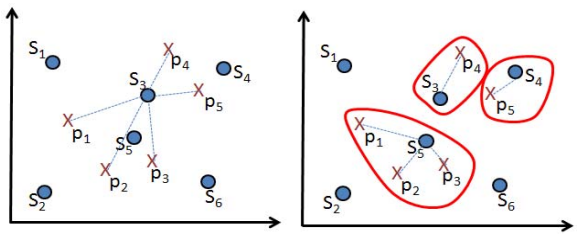

Fig. 3: Possible Environment States of the RUG Lab Dataset

sample readings x_h in V_g from the possible states that defines x_h, such that:

$$l_h = \arg \min_{1 \leq i \leq M} ||s_i - x_h|| \quad (3)$$

In Figure 3, samples x_1 to x_3 are nearby to state s_5; therefore, $l_1 = l_2 = l_3 = 5$. On the other hand, sample readings x_4 and x_5 are nearby to states s_3 and s_4, correspondingly; therefore, $l_4 = 3$ and $l_5 = 4$. An *accurate state identification* component determines the precise state a_g, e.g., the state that defines the entire set of sample readings in V_g that group mutually:

$$a_g = \arg \max_{1 \leq i \leq N} |\{x_h \in V_g | l_h = i\}| \quad (4)$$

The process explained so far uses a set of environment states $S = \{s_1, s_2, ..., s_N\}$ to define the likely physical status traversed by the environment and the system faults. The *environment state grouping* component provides an updated estimation of the states set. It means, if any ordinary deviations in the clustered states happen, the *environment state grouping* component uses an online statistical clustering algorithm on the given sample to capture them. The component uses the incoming sample readings set V_g to update the value of obtained states. The ultimate goal of this component is to provide a set of states that is obtained from given dataset and completely represent the sample.

B. Environment Modeling through Hidden Markov Model

The proposed approach examines the structural properties of two HMMs generated from the sample data to categorize the data and system faults affecting the WSN. Though, both hidden states and samples states imitate the likely environment states $\{s_1, s_2, ..., s_N\}$, these states are generated with the *environment state grouping* component. The proposed approach considers two HMMs: (1) an HMM which maps a set a_g (based on hidden/accurate changes of the atmosphere) onto a set v_g (based on the visible changes of the atmosphere); and (2) an HMM that maps a set a_g onto a system fault set f_g (based on the changes of a faulty sensor). Hidden/accurate states of the deployed sensors a_g are not directly visible. They are generated by an *accurate state* component and are, therefore, accessible when constructing the two types of HMMs. We use an on-line method to generate an HMM, which estimate the current hidden state of the model (a_g) and the current sample reading (v_g or f_g, based on the used HMM).

IV. Data and System Faults

Wireless sensor nodes experience two broad categories of faults, both affecting the performance. The first type is system faults, which naturally happes because of calibration, connection or hardware failures, communication failures or low battery states. The second type is data faults, where a sensor node performs normally apart from its sensing sample reading, leads to major biased or random faults, such as stuck-at, offset or gain. In general, a fault is defined as a variation from the probable model of the phenomenon, if the ground truth is available with high confidence. In this paper, we study faults from a data-centric perspective such as stuck-at, offset and gain, and from system-centric view such as calibration and low battery. It means that identification and classification among data and system fault types is necessary to have the chance of performing a recovery action. Next, we define most common data and system fault types for WSNs.

A. Data Fault Models

Wireless sensor nodes interact directly with the environment to measure physical attributes and, therefore, there is a high probability to a have a system fault which spoils them rapidly. Field studies [10], [15] specify that faults originating in a degraded sensor device are a major cause of unreliability in a wireless sensor network. We are interested to find a simple framework within which we can report as many faults as possible. We assume that the true measurement values come from a defined or random process, and noise or fault is added on that values through either an additive noise process or a linear deterministic function. Our generalized model for faults is then as follows. First, a fault-free value v is represented as follows:

$$x = v + \varepsilon$$

where v is the true value of a phenomenon and ε is an additive noise variable. Even, the most expensive system has some measurement noise in reality. We identify this reading as a fault-free value. When the sensor reading is faulty, we assume it represent the subsequent general form:

$$x' = \alpha_0 + \alpha_1 v + \varepsilon$$

where α_0 is the *offset* and α_1 is the *gain* values. We can define many data faults with this simple linear relationship among true phenomenon and fault. We give a simple taxonomy of three data faults for sensing devices.

1) Offset Fault

An *offset* fault is defined as a sudden deviation from the normal data with a constant amount. It usually exhibits itself as a *calibration offset*; an additive constant, which is added to the fault-free sample reading. It implies that the faulty reading is only based on the current sample reading and the current *offset*. The faulty reading is modeled by:

$$x' = \alpha_0 + v + \varepsilon$$

2) Gain Fault

The *gain* data fault is defined as the rate of change of the measured sample with respect to the expectations over an extended period of time. In the presence of a *gain* fault, faulty sensor sample readings are changed by a multiplicative constant, which is multiplied to the fault-free sample reading. It is hard to distinguish the *gain* fault from an *offset* fault without any ground truth and domain knowledge. The faulty reading is modeled by:

$$x' = \alpha_1 v + \varepsilon$$

3) Stuck-at fault

The stuck-at fault is defined as a series of sample readings that experience zero or roughly zero difference over a period of time greater than expected. An example is shown in Figure 4. The *stuck-at* fault shows a sensor stuck at a particular sample value. Frequently, this is a reading at the higher or lower boundary of the sensing range. We noticed in the real-world living lab dataset, that temperature sensors get stuck at value (122 °C), when the sensor has a low battery level. Usually, temperature sensors have a sensitivity range between -40 °C and 123.8 °C. The faulty reading is modeled as:

$$x' = \alpha_0 \quad .$$

B. System Faults

WSNs data faults are typically due to the following system faults: *calibration*, *low battery*, communication and connection/hardware failures. We provide a classification of system faults together with examples coming from real-world deployments. The goal of our work is to identify and classify a system fault that changes the expected performance of a system.

1) Calibration

Calibration problems can be a root cause of faulty data in many cases. Many papers cite the trouble in *calibration*, particularly while the sensor network is deployed [22], [23], [2], [10]. Usually, two different types of *calibration* faults can happen, e.g., *offset* and *gain* faults. Since these faults may be combined in numerous ways, these kinds of faults are difficult to handle without any expert or domain knowledge. Sometimes expert and domain knowledge is available but ground truth is not, yet it is hard to distinguish between a *calibration* fault and normal phenomenon variations. Usually, *calibration* data faults are defined relative to the ground truth. Suppose that the ground truth is not available then *calibration* data faults can only be determined relative to a probable fault model. The fault model is defined on the environmental context. Normally, spatial correlation is very important for defining the fault model when the ground truth is not available. We used HMM to identify *calibration* faults because we have ground truth information and expert knowledge.

2) Low Battery

A common cause for faulty data is a *low battery* e.g. [10], [8]. A low battery level determines how long a sensor will work and when a sensor value will start transmitting faulty values. From the real-world dataset, we plot in Figure 4 measured temperature readings and the battery voltage levels. The temperature sensor begins to fail at roughly the voltages representing that the failure is a probable. Once the battery voltage falls under such value, the temperature sensor readings then remain *stuck-at* one value for the rest of the operation. In [13], the authors conclude that sensor's battery failures are

responsible of most of the faults in the data. When the battery voltage level was less than 2.4V or greater than 3V, behavior similar to that of Figure 4 manifested itself. Sensor's battery supply affects the system performance by either adding noise or giving faulty data depending on the type of sensor and application.

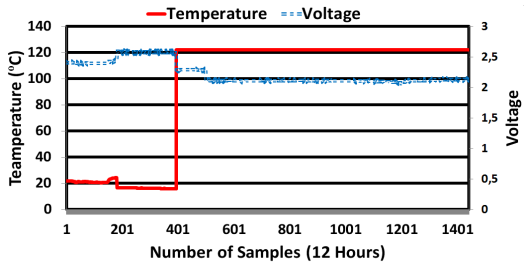

Fig. 4: Stuck-at Fault in real-world Dataset

V. Data and System Fault Classification

To identify and classify WSN faults, we build two mathematical models that capture the system's dynamic behavior, namely a HMM_{AV} linking accurate sensor states to visible sensor states, and a HMM_{AF} connecting accurate sensor states to data and system fault states. Existence of data and system faults change the scenario of the WSN deployment space. The Markov Models M_A is based on the array of the accurate sensor states a_g. The Markov Models M_V is based on the array of the visible sensor states v_g. Both HMM models consist of the same quantity of states and set of transitions even in case of presence of faults. We examine the columns and the rows of the C^{AV} and the HMM_{AV} matrices to see whether they are orthogonal or not.

$$\forall g,h : \sum_i c_{gi}^{av} c_{hi}^{av} = \delta_{gh} and \forall g,h : \sum_i c_{ig}^{av} c_{ih}^{av} = \delta_{gh} \quad (5)$$

The first equation states the condition that if two hidden states are distinct, then they generate two different sample symbols. The second equation states the condition that if two-sample symbols are distinct, then they are generated by two different hidden states. In this paper, hidden states of HMM_{AV} are defined as accurate environment states, and sample symbols of HMM_{AV} are defined as visible environment states. Additional classification of system and data fault types require a detailed structural analysis of the HMM_{AV} for system and the HMM_{AF} for data faults' type.

A. Data Fault Type Detection

The proposed method examines the HMM_{AF} model to identify the detected data fault type. The behavior of the data fault is examined as a function of the accurate state of the sensor by using the HMM_{AF}. The hidden states of HMM_{AF} is defined as accurate sensor states and HMM_{AF} model defines sample symbols as data/system faults.

Stuck-at: The stuck-at fault is defined as a faulty sensor continuously measuring the same value, which is usually beyond the possible state of the sensor. Consequently, all accurate sensor states are mapped onto the identical fault state. Formally, this corresponds to saying that the sample symbol probability distribution C^{AF} of HMM_{AF} is such that it has one column (y) that has all ones and other columns of all zeros:

$$\exists y : \forall g : c_{gh}^{af} = \begin{cases} 1 & \text{if } h = y \\ 0 & \text{if } h \neq y \end{cases} \quad (6)$$

Gain and Offset: A *gain* fault is characterized by a faulty sensor reporting a faulty value that changes accordingly with the accurate state of the sensor. It concludes that one-to-one mapping exists among accurate and fault states. This also holds true for an *offset* fault. The rows and columns from the matrix C^{AF} is orthogonal in gain fault. The same strategy is used for an *offset* fault.

$$\forall g,h : \sum_i c_{gi}^{af} c_{hi}^{af} = \sum_i c_{ig}^{af} c_{ih}^{af} = \delta_{gh}. \quad (7)$$

To further, categorize between *gain* and *offset* faults, we need to calculate the data features. For example, ratio and the difference between the attributes of corresponding accurate and data/system faults in HMM_{AF}. A *gain* fault usually manifests as a constant ratio, while an *offset* fault manifests as a constant difference. Assuming an accurate state $s^a = (x_1^a, x_2^a, ..., x_n^a)$ linked with a system fault state $s^f = (x_1^f, x_2^f, ..., x_n^f)$ in HMM_{AF} if there is a constant $K = (k_1, k_2, ..., k_n)$ such that $\forall g : \frac{x_g^a}{x_g^f} = k_g$ for a *gain* fault or $\forall g : x_g^a - x_g^f = k_g$ for an *offset* fault.

B. System Fault Detection

To determine the type of a detected system fault, we look at the HMM_{AV}. Through this model, we can study the consequence of the system fault on the visible state of the sensor as a function of the accurate states of the sensor. In the absence of data and system faults, each accurate state of the sensor corresponds to a single visible state of the sensor. One can conclude that prevalence of a system fault in WSN alters the one-to-one mapping.

Low battery: The *low battery* system fault is characterized by an accurate sensor state being associated with multiple sample sensor states (e.g., states g and h). In this case, columns x and y of matrix C^{AV} are not orthogonal: $\exists g,h : \sum_i c_{ig}^{av} c_{ih}^{av} \neq 0$.

Calibration: A *calibration* system fault is characterized by an accurate sensor state being associated with a single visible sensor state. In principle, the orthogonality of the matrix C^{AV} is not affected by the system fault. The attributes of the corresponding accurate and visible states of HMM_{AV} are examined to classify the fault as a *calibration* one. For example, in the presence of such a fault, an accurate state $s^a = (x_1^a, x_2^a, ..., x_n^a)$ connected with a visible state $s^v = (x_1^v, x_2^v, ..., x_n^v)$ in HMM_{AV} is such that $\forall g : x_g^a \neq x_g^v$.

VI. Experimental Results

To validate our proposal, we have used data collected from sensors deployed in an actual living lab realized in the context of an European Framework Seven project, in particular, pressure, PIR, acoustic, temperature, humidity, and light intensity

sensors. We examined the measurements collected every 10 seconds in 15 consecutive days at a base station. For the dataset, the ground truth is also available. The dataset is of medium size, consisting of slightly more than 48,600 samples. We call the dataset as RUG Lab (where RUG stands for Rijksuniversiteit Groningen).

In this section, we present our findings on the occurrence of data and system faults in temperature and humidity sensors samples by applying the HMM method to the given dataset. The dataset exhibited a mixture of *offset*, *gain* and *stuck-at* data faults because of system faults such as the *low battery* and *calibration* faults.

A. Data Fault Classification

The temperature change continuously during the day as visible in Figure 5. This observation remains true for the whole of the measurement period. The environment states grouping component needs an initial estimate for the set of model states. This early approximation is built on historic data or entirely randomly. We show results based on a preliminary set of six states obtained using an off-line clustering algorithm on the given dataset from the RUG Lab. The deployed sensors in the RUG Lab measures data every 10 seconds and transmits to the base station for analysis. Our window size is based on 360 samples, which is equivalent to one hour and gives enough time granularity and statistical meaning, e.g., mean, differences, ratios (about three hundred and sixty sensor readings in average).

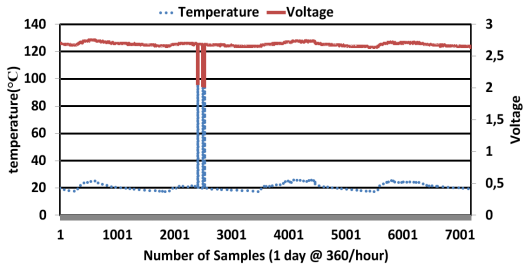

Fig. 5: Temperature Variation for one Full day

Figure 6 shows the accurate Markov Model M_A of the environment (temperature sensor), as expected by the process defined in the previous section. Five main possible states of the given scenario can be recognized, such as (13, 17, 19, 22, 24), from the given dataset and each state represent a temperature value. One extra state (15) results from variations within the sample; however, it is not considered as a key state of the system because it has low transition probability.

Figure 7 shows the two HMMs, that are HMM_{AV} and HMM_{AF} for sensor 3 learned from the RUG Lab dataset. The sample symbol probability matrix (C^{AV} and C^{AF}) and state transition probability matrix D are shown in Table I and Table II, respectively. We conclude that the rows and the columns of C^{AV} are almost orthogonal ($\sum_i c^{av}_{gi} c^{av}_{hi} < 0.1$ for $g \neq h$, and $\sum_i c^{av}_{gi} c^{av}_{hi} > 0.85$ for $g = h$), based on the relation described in Section III. The matrix C^{AF} is shown in Table II,

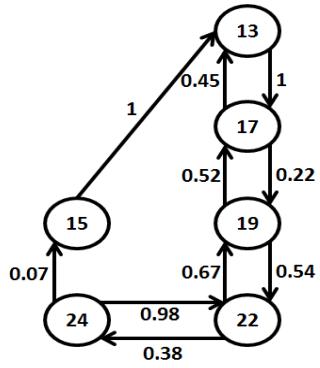

Fig. 6: Predicted/possible Markov Model of the temperature sensor

where we notice that one column is almost null and another column (state 122) is made almost exclusively of ones. This leads to properly classifying sensor 3 to be in a *stuck-at* state.

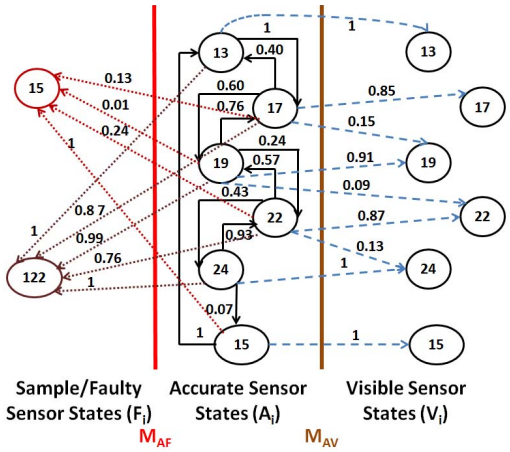

Fig. 7: HMMs for Faulty Sensor 3

$x\downarrow, y\rightarrow$	13	17	19	22	24	15
13	1	0	0	0	0	0
17	0	0.92	0.08	0	0	0
19	0	0	0.91	0.09	0	0
22	0	0	0	0.87	0.13	0
24	0	0	0	0	1	0
15	0	0	0	0	0	1

TABLE I: C^{AV} matrix for faulty sensor 3 - Stuck-at Fault

A similar investigation is done for the *offset* fault with sample obtained from sensor 4. Table III and Table IV show the resulting samples symbol probability matrices, C^{AV} and C^{AF}, respectively. The matrices are roughly orthogonal. Moreover, when calculating the ratios $\frac{x_i^a}{x_i^f}$ and the differences $x_i^a - x_i^f$ among the characteristics of corresponding accurate states. We calculate low variance (0.006), ratios with average (1.23), high variance (0) and differences with average (5). This leads us to correctly classify sensor 4 as affected by an *offset* fault.

$x\downarrow, y\rightarrow$	15	122
13	0	1
17	0.03	0.97
19	0.01	0.99
22	0	1
24	0	1
15	1	0

TABLE II: C^{AF} matrix for faulty sensor 3 - Stuck-at Fault

$x\downarrow, y\rightarrow$	13	17	22	27	32
13	1	0	0	0	0
17	0	0.8	0.2	0	0
22	0	0.02	0.98	0	0
27	0	0	0.001	0.999	0
32	0	0	0	0.001	0.999

TABLE III: C^{AV} matrix for faulty sensor 4 - Gain Fault

B. System Fault Classification

Next we evaluate the accuracy and robustness of the proposed system fault type detection method. We injected artificially system faults into the system setup under different system fault scenarios. Ideally, before inserting system faults into the deployed sensors, we should confirm that the real-world deployment does not have any faulty sensor. By injecting system faults, the faulty nodes change their behavior and turn the system into a new state. We injected artificial faults into one-fourth of the deployed sensors. As discussed in Section III, system faults are classified by investigating the samples symbol probability distribution, C^{AV}. Table V shows matrix C^{AV} for faulty sensor 6. The corresponding accurate states and visible states are different. This shows that *calibration* system faults do not influence the orthogonality of C^{AV}.

A similar artificial injection experiment can be done for the *low battery* system fault. For example, we replaced the batteries of a few sensors to put them in a low voltage level. The system fault (*low battery*) deletes correct sensor states (22) by reporting faulty temperature values, which is much higher than other sensor reported values. In the following, faulty nodes inject high temperature values into the system. As a result, visible states of the environment changed overall, while the accurate environmental temperature remains almost constant. Injected values are close to the maximum sensitivity range of the temperature sensor. Table VI shows sample probability matrix C^{AV} obtained by the explained approach for a faulty sensor 5. The column probabilities of the obtained matrix are not orthogonal (notice column 19 and 122). This shows that the *low battery* system fault has created an extra state (state 122).

$x\downarrow, y\rightarrow$	13	17	22	27	32
13	0	0	0	0	0
17	0	0	0.85	0	0
22	0	0.86	0	0	0
27	0	0	0.87	0	0
32	0	0	0	0.46	0

TABLE IV: C^{AF} matrix for faulty sensor 4 - Gain Fault

$x\downarrow, y\rightarrow$	13	17	19	22	24
13	0	0.87	0	0	0
17	0	0	0.92	0	0
19	0	0	0	0.94	0
22	0.87	0	0	0	0
24	0	0	0	0	0.86

TABLE V: C^{AV} matrix for Calibration system fault affected sensor 6

$x\downarrow, y\rightarrow$	17	22	13	19	122
17	1	0	0	0	0
22	0	1	0	0	0
13	0	0	1	0	0
19	0	0	0	0.31	0.69

TABLE VI: C^{AV} matrix for Low battery system fault affected sensor 5

VII. Related Work

Sensor measurements can deviate from their predictable values due to an unexpected event or without any known causes, particularly in the context of environmental monitoring. Two recent papers [17], [18], propose fault detection methods to detect data faults. However, neither of them focuses on data fault's causes. In [17], we present a hybrid fault detection approach which uses HMM to identify data faults, e.g., spikes, noise, outlier and stuck-at ones. The authors in [18] propose an approach to detect only short and constant data faults by using rule-based and estimation-based methods. Our fault detection method not only flags faulty measurements as faults but also identifies the type of faults and their causes, which helps to recover to correct operation.

HMMs have been extensively explored in fault detection systems [11], [12], [1], [5], [14]. In [1], the authors use a Markov chain to classify standard against inconsistent actions by considering diverse metrics. In [5], a HMM is learned to identify faults against web-based and web-servers applications. In [14], authors examine the accuracy of a Markov chain-based method and determine that Markov chains perform well in fault detection. HMMs present a better scientific utensil than basic Markov models.

In [19], the authors present an approach based on pattern recognition that is also joint with a finite-state HMM. The approach presents a beneficial technique for modeling temporal context in monitoring faults in complex dynamic systems. In [21], the authors use a HMMs strategy for intrusion detection, using distributed observation across multiple nodes. The authors of [9] present a novel dynamic, machine learning-based technique for automatically detecting faults in HVAC systems. In addition to dynamic Bayesian Networks and HMMs, data fusion is also used to combine fault detection results from multiple fault models in an attempt to achieve a more accurate fault detection outcome. The method in [9]

develops HMMs to learn probabilistic relationships between groups of points during both normal and faulty operation. HMMs are effectively used to anomaly detection as a method to model usual actions. Despite the above research effort, there does not yet exist a well-accepted method for detection of data and system faults and their classification in wireless sensor networks. A cutting edge challenge is to develop the capability to carry out fault diagnosis in terms of its identification and classification for data and system faults. We proposed an approach based on HMMs. Our approach not only detects both data and system faults, but also identifies their types. Nevertheless, the proposed approach mainly focuses on data faults occurred by *calibration* and *low battery* system faults.

VIII. Concluding Remarks

We presented a statistical approach to detect faults in wireless sensor networks. The proposed approach learns the possible system outcome dynamically without any distinct training period. Furthermore, it can be used to identify and classify data and system faults considering the structural relations between two kind of HMMs dynamically created. The focus of the present work lies on the calibration of data and system faults. We evaluate our proposed approach with real world data coming from the RUG Lab dataset. The approach can be extended to detect and classify more data fault types such as outlier and spikes ones and particularly for system fault types, such as communication failures and environment out of range faults. Our future work will focus on the extension of the framework to a larges set of fault types and a borader evaluation with actual datasets coming from physical installations.

Acknowledgement

The research is supported by the FP7 EU project GreenerBuildings, contract FP7-258888 and NWO project Energy Smart Offices, contract 647.000.004.

References

[1] S. Jha, K. Tan, and R. A. Maxion. *Markov Chains, Classifiers, and Intrusion Detection*, In Proceedings of the 14th IEEE workshop on Computer Security Foundations (CSFW 2001). IEEE Computer Society, Washington, DC, USA.

[2] L. Balzano and R. Nowak. *Blind calibration of sensor networks*. In Proceedings of the 6th international conference on Information Processing in Sensor Networks (IPSN 2007). ACM, New York, NY, USA, p. 79-88.

[3] L. Rabiner, *A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of IEEE, pages(257-286), 1989.

[4] *GreenerBuildings*, http://www.greenerbuildings.eu/, 2011.

[5] C. Kruegel and G. Vigna, *Anomaly detection of web-based attacks*, In Proceedings of the 10th ACM conference on Computer and communications security (CCS '03). ACM, New York, NY, USA, p. 251-261.

[6] C. Chong, and Srikanta P. Kumar. *Sensor networks: Evolution, opportunities, and challenges*, Proceedings of the IEEE, 2003, p. 1247-1256.

[7] C. Haowen, P. Adrian, P. Bartosz and S. Dawn, *SIA: Secure Information Aggregation in Sensor Networks*, Journal Computer Security 15, January 2007, p. 69-102, Amsterdam, The Netherlands.

[8] R. Szewczyk, J. Polastre, A.M. Mainwaring and D.E. Culler, *Lessons From A Sensor Network Expedition*, European Conference on Wireless Sensor Networks, 2004, p. 307-322.

[9] S. R West, Y. Guo and X R. Wang. *Automated Fault Detection And Diagnosis Of HVAC Subsystems Using Statistical Machine Learning*. 12th International Conference of the International Building Performance Simulation Association, 2011.

[10] N. Ramanathan, L. Balzano, M. Burt, D. Estrin, T. Harmon, C. Harvey, E. Kohler, S. Rothenberg and M. Srivastava, *Rapid deployment with confidence: Calibration and fault detection in environmental sensor networks*, Center for Embedded Networked Sensing, UCLA and Department of Civil and Environmental Engineering, MIT, TECH. Report 2006.

[11] C. Warrender, S. Forrest and B. Pearlmutter, *Detecting Intrusions Using System Calls: Alternative Data Models*, In IEEE Symposium on Security and Privacy, IEEE Computer Society, 1999, p. 133-145.

[12] C. Sung-Bae and H. Sang-Jun, *Two Sophisticated Techniques to Improve HMM-Based Intrusion Detection Systems*, Recent Advances in Intrusion Detection, LNCS Springer Berlin / Heidelberg, p. 207-219, 2003.

[13] G. Tolle, J. Polastre, R. Szewczyk, D. Culler, N. Turner, K. Tu, S. Burgess, T. Dawson, P. Buonadonna, D. Gay, and W. Hong. *A macroscope in the redwoods*. In Proceedings of the 3rd international conference on Embedded networked sensor systems (SenSys 2005). ACM, New York, NY, USA, p. 51-63.

[14] N. Ye and Y. Zhang and C. M. Borror, *Robustness of the Markov-chain model for Cyber-attack detection*, IEEE Transaction on Reliability, 2004, 53(1), p. 116-123.

[15] M. Alan, C. David, P. Joseph, S. Robert and A. John, *Wireless sensor networks for habitat monitoring*, In Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA '02). ACM, New York, NY, USA, p. 88-97.

[16] K. Ni, N. Ramanathan, M. N. H. Chehade, and M. Srivastava, *Sensor network data fault types*. ACM Trans. Sensor Network. 5, 3, Article 25 (June 2009), 29 pages.

[17] E. U. Warriach, K. Tei, T. A. Nguyen, and M. Aiello, *Fault detection in wireless sensor networks: a hybrid approach*, 11th ACM IPSN'12, New York, NY, USA, p. 87-88.

[18] A. Sharma, L. Golubchik, and R. Govindan, *On the prevalence of sensor faults in real-world deployments*. Proceedings of the IEEE Communications Society Conference on Sensor, Mesh and Ad-Hoc Communications and Networks (SECON) 2007.

[19] P. Smyth, *Hidden Markov models for fault detection in dynamic systems*, Pattern Recognition, Volume 27, Issue 1, January 1994, p. 149-164.

[20] I. Georgievski, V. Degeler, G. A. Pagani, T. A. Nguyen, A. Lazovik, and M. Aiello, *Optimizing Energy Costs for Offices Connected to the Smart Grid* , IEEE Transactions on Smart Grid, 2012. http://www.cs.rug.nl/ds/uploads/pubs/optimizingOffices.pdf.

[21] R. Khanna and H. Liu. *Control theoretic Approach to Intrusion detection using a Distributed Hidden Markov model*, IEEE Wireless Communications, vol.15, no.4, p.24-33, August 2008.

[22] P. Buonadonna, D. Gay, J.M. Hellerstein, W. Hong and S. Madden, *TASK: Sensor Network in a Box*, European Conference on Wireless Sensor Networks, 2005, p. 133-144.

[23] V. Bychkovskiy, S. Megerian, D. Estrin, and M. Potkonjak, *A collaborative approach to in-place sensor calibration*. In Proceedings of the 2nd international conference on Information processing in sensor networks (IPSN'03), Springer-Verlag, Berlin, Heidelberg, p. 301-316.