Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 83 (2016) 66 – 73

Procedia
Computer Science

The 7th International Conference on Ambient Systems, Networks and Technologies
(ANT 2016)

Distributed Fault Detection in Smart Spaces Based on Trust Management

Sila Ozen Guclu[a], Tanir Ozcelebi[a], Johan Lukkien[a]

[a] *Eindhoven University of Technology, Department of Mathematics and Computer Science, Den Dolech 2, 5600 MB Eindhoven, The Netherlands*

Abstract

Application performance in a smart space is affected by faulty behaviours of nodes and communication networks. Detection of faults helps diagnosis of problems and maintenance can be done to restore performance, for example, by replacing or reconfiguring faulty parts. Fault detection methods in the literature are too complex for typical low-resource devices and they do not perform well in detecting intermittent faults. We propose a fully distributed fault detection method that relies on evaluating statements about trustworthiness of aggregated data from neighbors. Given one or more trust statements that describe a fault-free state, the trustor node determines for each observation coming from the trustee whether it is an outlier or not. Several fault types can be explored using different trust statements whose parameters are assessed differently. The trustor subsequently captures the observation history of the trustee node in only two evidence variables using evidence update rules that give more weight to recent observations. The proposed method detects not only permanent faults but also intermittent faults with high accuracy and low false alarm rate.

Keywords: distributed fault detection; intermittent fault; outlier; opinion extraction; sensor nodes; trust; smart spaces; reliable; scalability

1. Introduction

In smart spaces, smart objects such as sensor nodes and smart phones are connected over a network in order to enable human-centric applications. By their very nature, smart spaces and smart objects therein are hampered by faulty behaviours of smart objects and communication networks interconnecting these. Furthermore, increment in number of devices also leads to higher failure probability. As this is a danger for the performance and even the correctness of smart space applications, faults must be detected and dealt with, preferably in an autonomous way. Dealing with faults in practice is often through physically replacing, rebooting and reconfiguring faulty parts by itself. If inevitable faults lead to (temporary) erroneous behaviors, a smart space and smart space applications must be able to quickly adapt and return back to a correct operation state without the constant need for human intervention. Fault

* Sila Ozen Guclu, Tanir Ozcelebi, Johan Lukkien
E-mail address: s.ozen@tue.nl, t.ozcelebi@tue.nl, j.j.lukkien@tue.nl

doi:10.1016/j.procs.2016.04.100

detection accuracy and precision are of huge importance as false alarms and false negatives come with a price, in the form of unecessary adaptations or failures. The focus of this paper is distributed fault detection. Adaptation is beyond the scope of this paper and is left as future work.

Faulty behavior is observed in the form of observations that highly deviate from the average, from here on called *outliers* and does not always take its source in a hardware or software fault. Short-term sequences of outliers can also be a result of fluctuations in the physical environmental conditions. Some examples are wireless signal strength loss due to people passing by, a packet loss due to temporary radio interference and corrupted data in magnetic storage due to cosmic rays. Some smart applications may tolerate short-term outliers without experiencing an application failure or a significant performance drop, while others may not. Obviously, what is 'short-term'must be defined per application. Making a good distinction between outliers that are mostly harmless, and faults that may lead to failures is crucial[7]. For this purpose, a typical approach in the literature is doing time series analysis of observations. This becomes especially challenging in distributed smart space architectures where most of the computation and decision making are done locally, without the computational resources and the global overview of a central server. This is because each smart object observes the environment from its own limited point of view.

Ideally, communication, memory and computation overheads of fault detection should be negligible and allow scaling to large networks[9]. A distributed fault detection method based on trust management is suitable for this purpose as the entire history of observations (information that is relevant) is captured in a single trust variable. A trust relationship between neighbors provides a local view of the network. There are two entities in such a trust relationship: a *trustor* and a *trustee*. The trustor relies on the trustee for the truth of a predicate. Bui et al.[4] states that such a predicate can be, for example, on the integrity, the security or the correctness of the data of the trustee. It can also be on the correct functioning of the trustee or the communication network in between. In the proposed method, the trustor tries to identify whether statements regarding the trustee given by predicates are trustworthy or not. Trust opinions are derived from subjective logic that represents specific belief calculus and considers uncertainty as given by Josang et al.[6].

Sensor nodes send their statistics to neighbor nodes and a base station. The trust is derived from the packet transmissions between sensor nodes. The trustee sends packets that consist of measurements such as communication statistics and sensor readings. Our fault model aims to identify sensor faults and communication faults which might be permanent or intermittent. A permanent fault means that the sensor node generates outliers continuously for the remainder of its lifetime. The source of intermittent faults is typically faulty hardware or faulty software. In this case the faulty component gives a series of outliers at irregular inter-arrival times[8]. Observations of smart objects are monitored for capturing the history of outliers, which may overall indicate a fault.

This paper presents a distributed method that detects both permanent faults and *intermittent faults* based on trust management. In comparison to approaches that employ time series analysis on a recent time-window[1,2,14], the proposed method has much lower memory footprint. It works with very little communication overhead as the only addition to network traffic is the exchange of a couple of variables between neighbor nodes. The entire history of observations is captured by using only two evidence variables, while giving more weight to the recent observations. Nevertheless, the proposed method differentiates outliers and faults with very high accuracy and high precision (or equivalently, low false alarm rate). In simple terms, after establishing which observations are outliers and which are not as we explain later, it works in the following way. Every *normal* observation builds up trust that there is no fault. Every outlier takes away trust, possibly at a different rate. We say that there is a fault if the trust value, whose initial value can be taken suitably depending on the application, falls below a certain threshold. This actually translates to 'there is not enough trust in the trust statement that describes the fault-free state'. Even if the trust value drops dramatically due to temporary outliers, trust can be re-established after observing a number of normal observations. On the other hand, permanent and intermittent faults give rise to a trend in which the dropping rate of trust due to outliers is on average faster than the trust reestablishment rate, resulting in a permanent distrust. Our proposed model investigates outliers in detail. Therefore, the proposed model could cause longer detection time in order to perform low false alarm rate. Mahapatro et al. stated that the number of tests and frequency of tests should be defined initially for detection of intermittent faults[10]. The proposed method tracks the history without the requirement of repetitive tests and parameters for the tests. Parameters of trust management only depend on the failure specification of the application. For example, flipping a bit in the payload of one packet out of every hundred packets is not a very significant problem for a video streaming application, while the same amount of bit flips (actually flipping even a single bit) would render a downloaded executable file unusable.

The paper is organized as follows: Section 2 reviews the related work on trust management and distributed fault detection. In Section 3, the trust management framework is presented. Section 4 describes the fault model and a case study. Section 5 provides the evaluation results. Finally, conclusions are drawn in Section 6.

2. Related Work

For distributed fault detection, Ding et al. proposed a method where each sensor node compares its own measurement with the median of measurements to identify whether it is faulty[11]. Voting among sensor nodes is a very well-known method for distributed fault detection. Two thresholds for spatial and temporal correlation are introduced and modified majority voting is employed for distributed fault detection by Chen et al[12]. In another work[13], if measurements of a sensor node deviate significantly from the weighted average, the sensor node is assumed faulty. The weighted average of the neighbors' measurements is calculated by considering the confidence levels of the neighbors.

Transient faults are detected by employing both spatial correlation (neighbor sensor nodes) and temporal correlation (over a time window)[14]. In this method, the length of the time window has to be kept very small for reducing the computational complexity to an acceptable level. Li et al. proposed an algorithm that deviates from measurement testing and fault detection[15]. In the algorithm, a local test is conducted to determine whether outliers are present.

Coordination in the clustered wireless sensor networks is employed for hierarchical trust calculation, where each sensor node calculates trust values for all its neighbors and then cluster heads aggregate trust values of each sensor node making a cluster overview[1,2]. Sun et al.[3] extend the trust model of Josang et al.[6] for fault-tolerant data aggregation by making use of both spatial and temporal correlation. It is also possible to go in the other direction and derive trust using fault detection and fault statistics. Xiao et al. stated that correlation between sensor readings can be analyzed for self-diagnosis in a distributed manner, which is then used to calculate trust[5]. If sensor node detects unusual data among its measurements, it enters to neighbor-diagnosis phase in order to determine its trustworthiness.

3. Proposed Method

We adopt the main stages of the trust management framework that is proposed by Bui et al.[4] and design each stage of the framework for fault detection. The framework describes the main stages of making a decision based on simple measurements. As shown in Fig. 2, it has five main stages: *observation*, *evidence*, *opinion*, *trust value* and *decision*. Observations about measurements are used as input to predefined evidence update rules, which lead to an opinion about the trust statement. A trust value is extracted from this, which in turn is used to make a decision. These stages are instantiated and clarified by the fault detection method that we propose in the rest of this section.

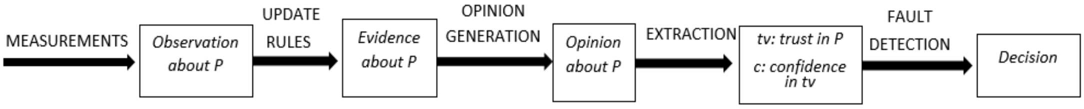


Fig. 1: Trust Management Framework

Decision, referring to a decision on *fault detection* in this work, is obtained based on trust. A trust statement[4] is given by *P(B, p, c, T)*, which means that the trustee B satisfies p in a given context c for a time interval T. An opinion, w_P^A, represents the belief, the disbelief and the uncertainity of a trustor A about the trust statement P.

3.1. Observation about Trust Statement

An observation is derived from the functions $s(\cdot)$ and $f(\cdot)$ that operate on $m_{i,t}$ denoting the measurement of a sensor node i at round t. In order to eliminate threshold dependency in computing $s(m_{i,t})$ and $f(m_{i,t})$, we use the modified

Thompson Tau statistical test for deciding whether a measurement is an outlier or not. Each sensor node collects measurements of its neighbors at each round. If the measurement of a sensor node highly deviates from the other measurements, the observation is identified as an outlier. Hence, spatially correlated sensor nodes are expected to observe similar measurements. In the modified Thompson Tau test, the set whose elements are the measurements of neighbors is analyzed. m_t and S_t are mean and standard deviation of this set at round t. The deviation of an individual data point, $\delta_{i,t}$, is calculated as follows:

$$\delta_{i,t} = |m_{i,t} - m_t| \tag{1}$$

The modified Thompson Tau test assumes that the measurements are normally distributed. In (2), $t_{\alpha/2}$ is the critical value of *student's t distribution* and n is the number of elements in the set (equivalently the number of neighbors). α is set as 0.05.

$$\tau = \frac{t_{\alpha/2}}{n(n-2+t^2_{\alpha/2})} \tag{2}$$

In the proposed method, $s(m_{i,t})$ and $f(m_{i,t})$ are given by

$$s(m_{i,t}) = \begin{cases} 0\ (outlier), & \delta_{i,t} > \tau S_t \\ 1\ (normal), & \delta_{i,t} \le \tau S_t \end{cases} \quad ; \qquad f(m_{i,t}) = 1 - s(m_{i,t}) \tag{3}$$

3.2. Evidence about a Trust Statement

As shown in Fig. 1, second aspect in the framework is the evidence whose update rules are defined for capturing observation history. Exponential Weighted Moving Average (EWMA) for computing the evidence has memory footprint that is considerably lower than time-window based approaches[1,2,14]. The evidence is represented by two variables; *se* and *fe*. The variable *se* is increased upon every *normal* observation ($s(m_{i,t}) = 1$), whereas *fe* is increased with every *outlier* ($f(m_{i,t}) = 1$). Note that those measurements, which fall into a critical value range and cannot be categorized as either normal or outlier add to uncertainity. These measurements translate into *unknown* observations that effectively weaken the evidence about the trust statement (both $se_{i,t}$ and $fe_{i,t}$). In EWMA, weights are assigned to observations such that an observation's contribution to the average decreases exponentially as it gets older. This gives higher weight to recent observations, while still taking into account the contributions of older observations. The smoothing factors α and β are the weights for the most recent normal observation and for the most recent outlier, respectively. Upon an unknown observation, $se_{i,t}$ and $fe_{i,t}$ are both scaled by γ. The formulas describing the evidence update rules are given in (4).

$$\begin{aligned} \text{Upon normal}: \ & se_{i,t} \leftarrow \alpha \cdot s(m_{i,t}) + (1-\alpha) \cdot se_{i,(t-1)} = \alpha + (1-\alpha) \cdot se_{i,(t-1)} \\ & fe_{i,t} \leftarrow \alpha \cdot f(m_{i,t}) + (1-\alpha) \cdot fe_{i,(t-1)} = (1-\alpha) \cdot fe_{i,(t-1)} \\ \text{Upon outlier}: \ & se_{i,t} \leftarrow \beta \cdot s(m_{i,t}) + (1-\beta) \cdot se_{i,(t-1)} = (1-\beta) \cdot se_{i,(t-1)} \\ & fe_{i,t} \leftarrow \beta \cdot f(m_{i,t}) + (1-\beta) \cdot fe_{i,(t-1)} = \beta + (1-\beta) \cdot fe_{i,(t-1)} \\ \text{Outcome unknown}: \ & se_{i,t} \leftarrow \gamma \cdot se_{i,(t-1)} \ \ and \ \ fe_{i,t} \leftarrow \gamma \cdot fe_{i,(t-1)} \end{aligned} \tag{4}$$

Note that the value of α determines the time required for trust (re-)establishment and is especially critical in the case of intermittent faults. If a long duration of trust establishment is desirable, the weight of the recent observation should be low, i.e. α should be close to 0. β determines the tolerance of the system to outliers. If only one outlier indicates a fault in the system, β is initially assigned close to 1. Otherwise, β is set to a lower value, meaning that a number of outliers back to back are required before detecting a fault.

3.3. Opinion about Trust Statement and Trust Value

The opinion w^A_P is given by a quadruple (b, d, u, a). The input variables to this function b (belief), d (disbelief) and u (uncertainity) are obtained from evidences, $se_{B,t}$ and $fe_{B,t}$ of trustee B. The formulas for opinion extraction, calculation of trust value ($tv(w)$) and confidence ($c(w)$) derived from this opinion are given by (5). Confidence represents how certain the trustor about the trust value.

$$\begin{aligned} b &= \frac{se_{B,t}}{se_{B,t} + fe_{B,t} + \epsilon}\ ,\ d = \frac{fe_{B,t}}{se_{B,t} + fe_{B,t} + \epsilon}\ ,\ u = \frac{\epsilon}{se_{B,t} + fe_{B,t} + \epsilon} \\ tv(w) &= b + u \cdot a\ ,\ \ c(w) = 1 - u \end{aligned} \tag{5}$$

Here, ϵ is a static initialization parameter in subjective logic that *i)* gives $u = 1$ (maximum uncertainity) initially and *ii)* ensures the equality $b + d + u = 1$ always holds, i.e. no division for $tv(w)$ by zero when there is no evidence yet. The weight factor a (base rate) determines the trust taken from uncertainity. In the proposed method, each trust statement describing a fault-free state is associated with a pre-defined threshold based on the fault specification of the system. If the trust value is below the threshold ($0 \le tv(w) \le 1$), a fault is detected by the trustor.

4. Case Study

4.1. Fault Model and Fault Types

A fault can be permanent or intermittent. Permanent faults are trivial to detect in the long run, although achieving a reasonable speed of detection while not introducing false positives is still a challenge for the fault detection method. Intermittent faults are more difficult to deal with, where the outliers resulting from the fault are observed in repetitive but not necessarily periodic bursts that are separated by random numbers of normal observations. Methods based on the time-window approach in the literature struggle especially for two reasons; *i)* determining a good window size is difficult, and *ii)* computational complexity and memory requirements become very high for large time windows. Consider the situation that sensor nodes follow duty cycling and send a data packet to their neighbors at each round. A sensor node may generate errors in sensor readings or die (sensor fault). A sensor node may also struggle to communicate (communication fault). At each round, two types of measurements are made for the fault detection: *sensor readings* and *packet reception*.

4.2. Trust Statements and Expected Trust Curves

Based on the considered fault, that includes intermittent and permanent faults in Fig. 2, three trust statements P1, P2 and P3 are defined regarding the trustor wireless sensor node S1 and the trustee wireless sensor node S2. P1, P2 and P3 collectively describe the fault-free state as given in Table 1.

Table 1: Trust Statements in Case Study

	Predicate	Measurement	Observation
P1	P1.p=‘S1 receives one packet from S2 at each round’	Packet reception	Whether the packet is received by the trustor
P2	P2.p=‘S2 is alive’	Packet reception	Whether the packet is received by the trustor
P3	P3.p=‘The measurements of S2 are accurate’	Sensor measurements	Whether the measurement is an outlier

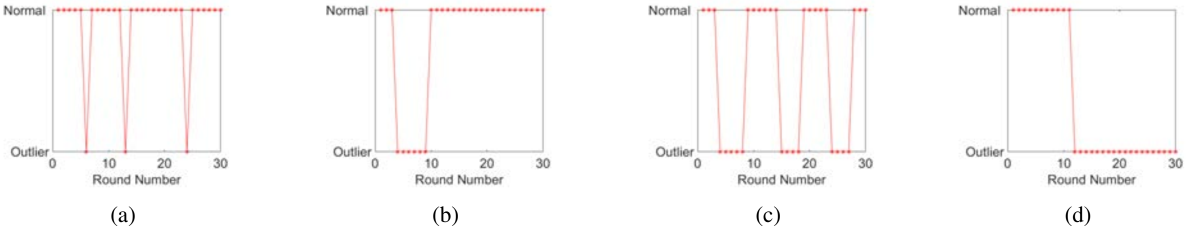
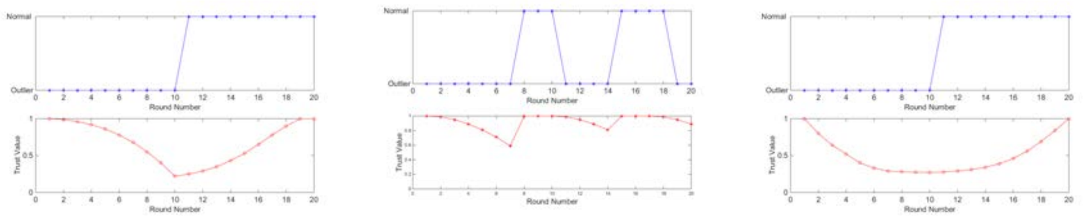


Fig. 2: The operation (a) with point outliers, (b) with a burst of outliers, (c) with intermittent bursts of outliers, (d) permanent fault. Note that in general (a), (b) and (c) may or may not indicate a fault depending on the fault specification of the system and the corresponding trust parameters α, β and γ. In this specific case study γ does not have any effect as there are no unknown observations.

Note that there is no universal trust establishment scheme for all trust statements. Logical trust establishment strategies have to be chosen. For example, if S1 receives only one packet from S2, this implies S2 is alive and it is logical that the trust of S1 in P2 is fully re-established instantenously. Establishing trust in P1, on the other hand, should require time because receiving only one packet (a single normal observation) does not imply receiving a packet consistently at each round. With the same reasoning, the trust in P3 should be established gradually as a single accurate measurement does not necessarily imply an accurate sensor. However, the strategy for degrading trust upon outliers for P1 and P3 should be different. This is because we assume that packet loss is common in a decent wireless network while inaccurate measurements (outliers) are not very common for a decent sensor. Thus, losing the first packet after a long sequence of good packet receptions can be due to an environmental factor such as a human passing by the sensor. An inaccurate measurement, on the other hand, is typically a sign of hardware problems. In general, the trust value should decrease and increase depending on the tolerance of the system to outliers and the trust taken from normal observations, respectively. The strategies that we consider logical for trust value adjustment for P1, P2 and P3 are shown in Fig. 3. The trustor assumes that the trustee is not faulty and trust value is initially set to 1. Alternatively, it could also set to 0.5 for representing a neutral view of the trustee node.



(a) P1 (b) P2 (c) P3

Fig. 3: Expected trust values for P1, P2 and P3

In P1 and P2, modified Thompson Tau test to identify outliers is not necessary due to binary measurements. Unknown observations are not considered in this use case. As a result, the sum of belief and disbelief is 1 at each round. Therefore, trust value is equal to $se_{i,t}$ in this case study. The minimum number of outliers that indicates a faulty state determines the value of β and is easily computed for permanent faults. In our previous work[7], we proposed a method for determining outlier patterns that imply a permanent fault. If n and T_{thr} are the minimum number of outliers, that determines the boundary of normal state, and the trust value threshold respectively. If n number of outliers is observed, it is considered as an indicator of a fault in the smart space application. β is given by

$$(1-\beta)^n = T_{thr} \quad \rightarrow \quad \beta = 1 - \sqrt[n]{T_{thr}} \tag{6}$$

5. Evaluation Results

Firstly, fluctuations of the trust value against intermittent faults are tested. We consider four temperature sensors that are neighbors of each other. One of the neighbors generates faulty temperature readings intermittently. In Fig. 4, the data generated by the faulty sensor node and the trustee from the perspective of three neighbors are shown by corresponding trust curves that are regarding trust statement, P3. The range of normal temperature readings is between *23°C* and *28°C*. On the other hand, the values of outlier data vary from *2°C* to *12°C*. The faulty sensor node generates outliers due to intermittent fault and point outliers at the same time as seen in data points. The other neighbors generate only point outliers. At each round whose duration is *1s*, each neighbor identifies outliers generated by its neighbors using the modified Thompson Tau test. In Fig. 4, trust value fluctuations of the trustee node against an intermittent faults are shown. The initial trust value is set to 0.5, representing a neutral view. A faulty sensor node generates a burst intermittent faults and point outliers. After the first intermittent burst of outliers (due to fault), the trust values drop dramatically for all trustors. With the following intermittent bursts of outliers, the trust values continue to decrease and they eventually go below the threshold as shown in Fig. 4. Even though the trust values decrease also due to point outliers, trust after a point outlier is re-established quickly thanks to the normal observations that follow. The trust curves of the three neighbors are not identical as point outliers are generated by all nodes.

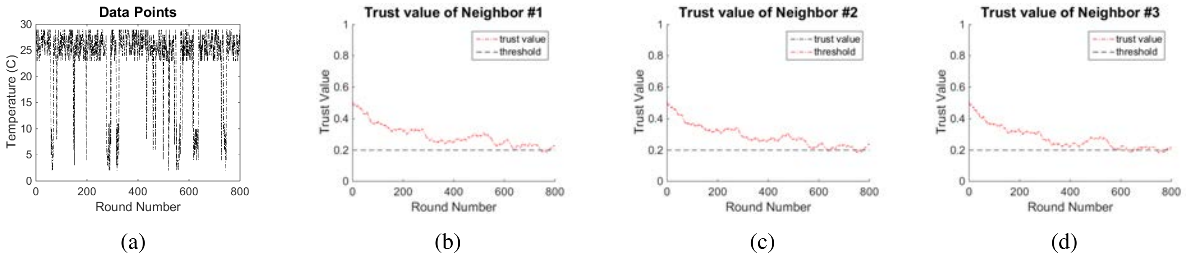


Fig. 4: (a)Data Points, (b)(c)(d) trust curves of three neighbors(trustors) of trustee(faulty node). All neighbors detect the intermittent fault.

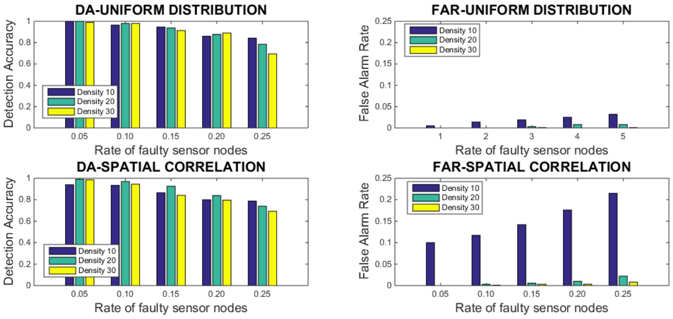


Fig. 5: Performance for different ratios of faulty sensor nodes

The detection accuracy (DA) and the false alarm rate (FAR) of the proposed method are tested using an event-driven simulator of temperature sensor nodes deployed in a *60m*×*60m* area on a square grid, each with a *6m* transmission range. DA is the ratio of number of detected faulty sensor nodes to the number of faulty sensor nodes, whereas FAR is the ratio of number of non-faulty sensor nodes, that are detected as faulty, to the number of non-faulty nodes. Sensor nodes evaluate the trust statements based on overhearing the packets without any communication overhead and calculate trust values for all three trust statements and for all their neighbors. In different tests, the network density is scaled by 10, 20 and 30. The corresponding numbers of sensor nodes in a simulation varies from 100 to 300 for different density levels.

In simulation, intermittent faults are injected the network for evaluating the performance of trust statement P3. The inter-arrival time of intermittent faults is given by the Poisson distribution. The duration of intermittent faults varies uniformly from *10s* to *20s*. Point outliers are also injected to the network in order to test performance of the proposed model in distinguishing between outliers and failures. An intermittent fault generates intermittent outlier bursts in sensor readings with Poisson distribution with mean *0.1 sample/min*. *Initial trust value*, T_{thr}, α and β are taken as 0.5, 0.2, 0.0005 and 0.001 respectively. α is considerably smaller than β, because the rate of trust increment should be slower than the rate of trust decrease. The performance is shown in Fig. 5.

There are two cases for the selection of faulty sensor nodes in the network: uniform distribution and spatial correlation. In uniformly distributed case, faulty sensor nodes are selected randomly and there is no spatial correlation among them. On the contrary, faulty sensor nodes tend to be located close to each other in spatial correlation case.

In Fig. 5, the ratio of faulty sensor nodes are changed between 0.05 and 0.25. The proposed method concentrates on trustworthiness of the data from the neighbors. Therefore, DA decreases gradually for all three density levels, when the ratio of faulty sensor node increases. The probability of the fault in the neighbors increases with increasing ratios. The lowest density network has the worst FAR performance compared with other density levels. Sensor nodes can not detect faulty sensor nodes accurately with a density of 10 due to scarcity of neighbors, making it difficult to identify outliers. The modified Thompson Tau test makes an assumption that data is normally distributed. Therefore, outliers are detected more accurately for higher densities (close to 7 neighbors for a density of 30), decreasing FAR.

In uniformly distributed faulty sensor nodes, the probability of multiple faulty sensor nodes in the same neighbor set is low. Hence, the performance in case of uniform distribution is better than in case of spatial correlation. FAR performance for spatial correlation is degraded especially for density 10 due to fewer neighbors. However, our proposed method could detect faults for spatial correlation with approximately 0.9 accuracy in lower faulty sensor node rates. It could detect not only individual faulty sensor nodes but also faulty region that contains several faulty sensor nodes. Neighbors of the faulty region could detect the faulty sensor nodes.

6. Conclusion

We have introduced a distributed fault detection method that has no communication overhead, requires very little memory space and is applicable to large-scale smart spaces. Sensor nodes tend to send their measurements to the base station for each pre-determined period. When a sensor node sends a packet, all of its neighbors receive the packet due to overhearing. Therefore, our proposed model does not cause any communication overhead. Trust relationships between sensor nodes are employed to detect intermittent and permanent faults. These trust statements are defined based on the fault specification and they describe a fault-free state. The observation history is captured in two evidence variables that are easy to work with in comparison to a time-window. A fault is detected when the trust value goes below a predefined threshold. The simulation results show that the proposed method can detect faults with high accuracy regardless of the distribution of faulty sensors (uniform or non-uniform) in the network. As expected, the detection accuracy decreases for higher ratios of faulty sensor nodes. The research on self-healing mechanisms in order to prevent faults or postpone the effects of faults in smart spaces is left as future work.

7. Acknowledgment

This work has been supported by the ProHeal project (n. 10017751) funded by ITEA2.

References

1. Shaikh, R.A.,Jameel, H.,d'Auriol, B.J., Lee, H. Lee,S. Song,Y., 2009. Group-Based Trust Management Scheme for Clustered Wireless Sensor Networks. IEEE Transactions on Parallel and Distributed Systems 20, 1698-1712.
2. Li, X.,Zhou,F., Du, J., 2013. LDTS: A Lightweight and Dependable Trust System for Clustered Wireless Sensor Networks. IEEE Transactions on Information Forensics and Security 8, 924-935.
3. Sun, Y., Luo, H., Das, S.K, 2012. A Trust-Based Framework for Fault-Tolerant Data Aggregation in Wireless Multimedia Sensor Networks. IEEE Transactions on Dependable and Secure Computing 9, 785-797.
4. Bui, V., Verhoeven, R., Lukkien, J., 2014. Evaluating Trustworthiness through Monitoring: The Foot, the Horse and the Elephant, Trust and Trustworthy Computing. Springer International Publishing 8564, 188-205.
5. Xiao, X., Peng, W., Hung, C., Lee, W., 2007. Using Sensorranks for In-network Detection of Faulty Readings in Wireless Sensor Networks. In: Proceedings of the 6th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE). ACM, China, pp. 1-8.
6. Jøsang, A., Hayward, R., Pope, S., Trust Network Analysis with Subjective Logic. In: Proceedings of the 29th Australasian Computer Science Conference(ACSC). ACM, Darlinghurst, Australia, vol. 48, pp. 85-94.
7. Ozen Guclu, S., Warriach, E.U., Ozcelebi, T., Lukkien, J., 2016. Improving Failure Prediction Accuracy in Smart Environments. In: Proceedings of IEEE International Conference on Consumer Electronics(ICCE). IEEE, Nevada, USA.
8. Bondavalli, A., Chiaradonna, S., Di Giandomenico, F., Grandoni, F., 2000. Threshold-based mechanisms to discriminate transient from intermittent faults. IEEE Transactions on Computers 49, 230 - 245.
9. Mahapatro, A., Khilar, P.M., 2013. Fault Diagnosis in Wireless Sensor Networks: A Survey. IEEE Communications Surveys Tutorials 15, 2000-2026.
10. Mahapatro, A., Panda, A., 2014. Choice of Detection Parameters on Fault Detection in Wireless Sensor Networks: A Multiobjective Optimization Approach, Wireless Personal Communications. Springer US 78, 649-669.
11. Ding, M., Chen, D., Xing, K., Cheng, X., 2005. Localized fault-tolerant event boundary detection in sensor networks. In: Prooceding of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM). IEEE, Miami, USA, vol. 2, pp. 902-913.
12. Chen, J., Kher, S., Somani, A., 2006. Distributed Fault Detection of Wireless Sensor Networks. In: Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks. ACM,Los Angeles, CA, USA, pp. 65-72.
13. Ji,S., Shen-fang, Y., Ma, T., Tan, C., Distributed, 2010. Fault Detection for Wireless Sensor Based on Weighted Average. In: Proceedings of 2010 Second International Conference on Networks Security Wireless Communications and Trusted Computing (NSWCTC). IEEE, Wuhan, China, vol. 1, pp. 57-60.
14. Lee, M., Choi, Y., 2008. Fault detection of wireless sensor networks. Elsevier Computer Communications 31, 3469 - 3475.
15. Li, W., Bassi, F., Dardari, D., Kieffer, M., Pasolini, 2015. G.,Low-complexity distributed fault detection for wireless sensor networks. In: Proceedings of 2015 International Conference on Communications (ICC). IEEE, UK, pp. 3469 - 3475.