DELIVERABLE D3.4 DESIGN OF TRANSIENT SENSOR ANOMALY DETECTION MACHINE LEARNING ALGORITHM

Identifier:	Deliverable
Class:	Report
Version:	
Version Date:	
Distribution:	Consortium Confidential
Responsible Partner:	Seoul National University
Contributors:	Seoul National University

EXECUTIVE SUMMARY

This deliverable describes the sensor anomaly detection algorithm based on Deep Neural Network anomaly detection. We describe the motivation, method design, and evaluation results in detail.

TABLE OF CONTENT

1 INTRODUCTION

Sensor-enabled smart home applications (e.g., energy management, security, and healthcare) are quickly emerging with the proliferation of low-cost sensors. Despite the opportunities, recent works have found that the sensor systems suffer from various types of sensor failures [7], including network failure, hardware malfunction, and incorrect sensory readings. In particular, detection of the contextual failures (sensors reporting incorrect values temporarily) is nontrivial, complicating the reliable operation of sensor systems. For example, widely used passive-infrared (PIR) motion sensors can report erroneous values due to temporary sunlight, which may wrongly control the appliances.

Accurate error detection for sensor systems remains an unsolved problem after many years of research. Many commercial sensor systems still rely on manual diagnosis, i.e., replacing/debugging sensors regularly or upon failure, which incurs high maintenance costs and extended downtime. To address the challenge, automated anomaly detection methods have been proposed [8, 10, 11, 17]. These methods train rules or machine learning models with long-term operational data to distinguish normal sensory readings from abnormal ones. However, prior works have two limitations. First, they focus on limited failure types; in particular, sporadic contextual failures have not been their primary interests. Second, these methods are limited to detecting a single sensor failure, which hardly handles simultaneous sensor faults (e.g., multiple motion sensors simultaneously affected by sunlight).

In this paper, we aim to explore the possibility of using Deep Neural Network (DNN) to identify simultaneous sporadic sensor anomalies. In particular, we propose an anomaly detection algorithm by extending Hypershpere Classification (HSC) [14], designed for one-class anomaly detection (i.e., normal vs. abnormal). Our algorithm models the normality of multi-sensor data with a small number of anomaly data. Unlike typical classification algorithms, it estimates the boundaries of normal multi-sensor data more reliably only with a minimal amount of per-sensor anomaly data. In addition, it identifies the anomaly associated with individual sensors instead of reporting the system failure as a whole.

We evaluate our method on a public smart home dataset. The detection performances for simultaneous sporadic sensor failure of the baselines drop up to 54.4% while ours drops up to 1.1%.

2 RELATED WORK

Table 1: Sensor Anomaly Detection System

System	Method type	Method	Target sensor failure types	Simultaneous sensor failure
Ours	Supervised Outlier Exposure	Hypersphere Classification	fail-stop, sporadic	✓
Idea [8]	Unsupervised learning	Association Rule Learning	fail-stop	✗
DETECTIF [10]	Unsupervised learning	Association Rule Learning	fail-stop, sporadic	✗
FailureSense [11]	Unsupervised learning	Gaussian Mixture Model	fail-stop	✗
CLEAN [17]	Unsupervised learning	Clustering	a single abnormal event	✗

We summarize prior sensor anomaly detection methods in Table 1. These methods take an unsupervised learning approach, which learns the pattern of normal sensor data. At runtime, these methods determine anomaly if the input data goes beyond the discovered normal data pattern. However, these works are limited to detecting fail-stop failures of a single sensor.

Association Rule Learning (ARL) is the most widely used method for smart home anomaly detection [2, 8, 10]. In the training stage, this method extracts association rules composed of antecedent and consequent sensors; the consequent sensor is determined based on the pattern of the antecedents. This method has a limitation in that the rules hardly represent rich features such as the number of sensor events and the existence of sensor events, making it challenging to identify complex patterns of contextual failures. Additionally, ARL detects the sensor anomaly based on the assumption that all antecedents follow normal behavior, which is invalid when multiple sensors simultaneously fail.

FailureSense [11] utilizes Gaussian Mixture Model (GMM) to learn the distribution of normal sensor data. FailureSense trains GMMs for each sensor to detect sensor-wise anomalies. GMM can detect non-fail-stop sensor failures. However, FailureSense does not reflect the relationship between multiple sensors.

CLEAN [17] proposes a clustering-based sensor anomaly detection algorithm. This work defines a distance between sensor events and cluster sensor events using a clustering algorithm [6]. At runtime, a sensor event is classified as anomalous when the distance between the event and the clusters is larger than the threshold. CLEAN only detects specific anomalous events but does not detect sporadic anomalies.

3 PROBLEM FORMULATION

In this section, we formulate a sensor anomaly detection problem.

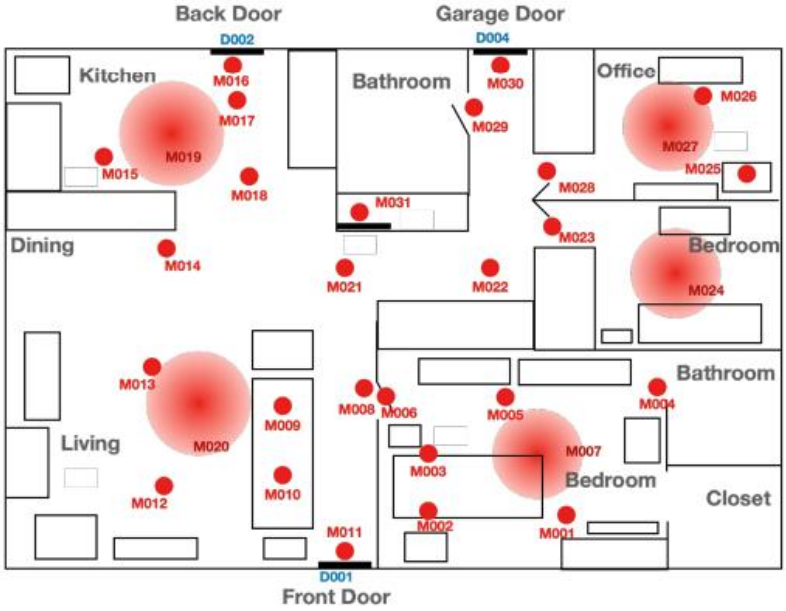

Figure 1: Floorplan of Aruba testbed of CASAS smart home dataset. The PIR motion sensor starts with M, and the door sensor starts with D. Red dots refer to the PIR motion sensors, and larger red circles refer to the long-range PIR motion sensors. There are three door-open sensors.

3.1 Smart Home Sensor Data

Our formulation assumes a set of binary sensors (e.g., a motion sensor triggers ON/OFF events). In particular, we assume a set of motion sensors deployed in a smart home (See Figure 1 from CASAS dataset [3]). The motion sensor passively measures infrared light in their field of view and triggers ON events when the change of infrared radiation amount exceeds the pre-defined threshold. It triggers OFF events when the amount of infrared radiation drops more than the threshold [5].

We formulate an event triggered by a motion sensor as $e_i =< t_i, s_i, f_i >$ where i is the index of the event, t_i is the timestamp, si is the triggered sensor ($s_i \in S$ where S is the set of sensors), and f_i is the event type where $f_i \in \{0,1\}$ (i.e., $f = 1$: ON event, $f = 0$: OFF event). Each ON event is followed by its corresponding Figure 1: Floorplan of Aruba testbed of CASAS smart home dataset. The PIR motion sensor starts with M, and the door sensor starts with D. Red dots refer to the PIR motion sensors, and larger red circles refer to the long-range PIR motion sensors. There are three door-open sensors. OFF event. One ON/OFF event pair of sensor s is composed of two events, $< t_k, s, 1 >$ and $< t_l, s, 0 >$, where tl is the minimum t_i of all $< t_i, s, 0 > (t_k < t_i)$. We assume sensor events are segmented into sensor data segments $1m = \{e_k, e_{k+1}, \dots, e_{k+g-1}\}$ by segmentation algorithms [4, 9, 12] at the start and end of motions.

3.2 Sensor Failure Types

We model two types of sensor failure: fail-stop failure, and sporadic failure. These two failures can both occur in a single sensor and multiple sensors simultaneously. The followings are details of two error types.

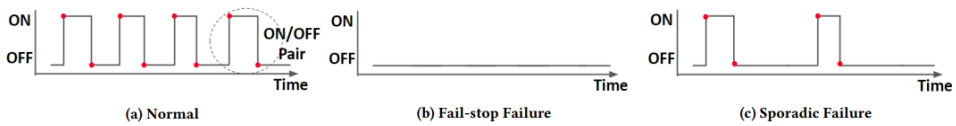

Figure 2: Sensor Failure Types. The red dots are the events triggered by sensors.

Fail-stop failure. This error indicates hardware failure due to aging, circuit short, or battery drain. When it occurs, the sensor no longer records nor sends data. In order to reproduce this error for evaluation, all events $e_i =< t_i, s_i, f_i >$ of a failed sensor s are discarded from the time t_i onward.

Sporadic failure. Sensor hardware is not broken, but it misses some ON/OFF event pairs for a certain duration. This failure occurs when the sensor's capacity (e.g., sensitivity, sensing range) is weakened due

to environmental changes (e.g., illumination, temperature, or humidity), temporary occlusion by objects (e.g., box, pet, or user), and the aging of the sensor. In such cases, partial data are reported normally while the remaining are missed. In evaluation, we reproduce sporadic failure by randomly discarding the ON/OFF event pairs of sensor s with the probability of ρ.

Importantly, these two sensor failure models extend to multiple sensors. That is, the above failures can occur in more than one sensor simultaneously. It's easy to imagine a case of simultaneous fail-stop failures on multiple sensors; just a few of the smart home's sensors are broken. A group of sensors also suffer from sporadic failure. For example, when strong afternoon sunlight comes through the windows to the living room, the smart home environment changes rapidly, and then the sensing range of some sensors is shortened. In evaluation, to reproduce this scenario, we inject sporadic failure with ρ=0.5 to three motion sensors (M009, M013, M020) in the living room of Figure 1.

3.3 Sensor Anomaly Detection

Sensor anomaly detection methods learn the normal pattern of sensor data with the training data and detect anomaly sensors at runtime. We build an anomaly detection model for each sensor. For each model, we denote the anomaly detection target sensor as the target sensor and the other sensors as non-target sensors. Each model is trained with the set of normal segments $\boldsymbol{M} = \{m_1^{norm}, \dots, m_M^{norm}\}$. Optionally, abnormal segments can be used for training. In the runtime, each model outputs an anomaly value as for sensor s where $a_s \in \{0,1\}$ with a_s = 0 for normal sensor, and a_s = 1 for the abnormal sensor.

4 APPROACH

In this section, we introduce our anomaly detection pipeline designed for simultaneous sporadic sensor errors.

4.1 DNN Method Selection

We first choose the proper DNN-based method for our anomaly detection problem. The prior methods learn the pattern of the target sensor with the given condition of non-target sensor data [8, 10, 11, 17]. These methods lack the ability to detect simultaneous faults because they have an invalid assumption that the non-target sensor data is normal. For example, ARL-based methods infer the anomaly state based on the assumption that all antecedent events are from normal sensors. Multiple abnormal data in the antecedent events incur low anomaly detection accuracy.

We apply a DNN algorithm to overcome the limitation of the prior works because DNN algorithms do not assume the normality of the non-target sensor data. There are a few DNN algorithm that fits our problem. DNN-based binary classification can be used as an anomaly detection method trained with normal and anomaly data. However, it is not trivial to collect a large amount of anomalous sensor data of various types. Unsupervised deep one-class classification methods are proposed to overcome the limitation of DNN-based classification, which learns a pattern of normal data in an unsupervised manner. For instance, Deep SVDD [13] learns a transformation to the feature space that the normal data is bounded in a small area. However, it has a shortcoming to learning sufficient features only with normal data, decreasing the anomaly classification accuracy.

We find that the DNN-based Supervised Outlier Exposure (OE) [14] method best fits our system. Supervised OE methods differ from unsupervised learning that they train the models with a small amount of anomaly data and differ from supervised learning that they assume that only normal data has a specific characteristic while anomaly data does not. Specifically, we adopt Hypersphere Classification HSC [14], which concentrates normal data around the origin, and pushes abnormal data far away from the origin on the feature space F. This desensitizes the anomaly detection accuracy to the threshold. Let ϕ (·|W): X → F be a neural network of HSC and $l(\cdot)$: F → [0, 1] be a function that maps the output to a probabilistic score. The objective of HSC is given as follows:

$$\min_{\mathcal{W}} \frac{1}{n} \sum_{i=1}^{M} y_i l(x_i) - (1 - y_i) \log(1 - \exp(-l(x_i)))$$

$$l(x_i) = \sqrt{\| \phi(x_i|\mathcal{W}) \|^2 + 1} - 1$$

where $y_i \in \{0,1\}$ is an anomaly label, with y_i = 0 denoting normal data, and y_i = 1 abnormal data. For each input x, HSC outputs the anomaly score as $s(x) = \left|\left|\phi(x|W)\right|\right|^2$. HSC compares the anomaly score with the threshold to detect an anomaly.

4.2 Simultaneous Sporadic Sensor Anomaly Detection

We extend the HSC algorithm for our anomalous sensor detection problem. We first need to convert the segments of sensor data mj to the vectors xj of the same size to be used as HSC input. We use two segment-wise values, the first timestamp of the first event in the segment, the time length of the segment, and two sensor-wise values, the number and the sum of the duration of ON/OFF event pairs, which makes the vector size 2|S| + 2. Then, we can train an HSC model with the training dataset {($x1$, $y1$), . . . , (xM, yM)}, where $yi \in$ {0, 1} is an anomaly label of xi with y = 1 denoting normal data and y = 0 abnormal data.

To enable sensor-wise anomaly detection, we train HSC for each sensor $s \in$ S with the normal data X $norm$ and abnormal data X $abnormal$ s of the sensor respectively. We find that the naive training method incurs a detection accuracy drop. When the data from non-target sensor s' ∈ S − s is abnormal, the HSC for sensor s anomaly detection classifies the data as abnormal even when s is normal. To handle this problem, we modify the normal data for training an HSC to detect sensor s anomaly as follows:

$$\mathbf{X}_s^{norm} = \mathbf{X}^{norm} \cup \bigcup_{s' \in \mathcal{S}-s} X_{s'}^{abnormal}$$

The neural network of HSC for each sensor ϕ (·|W) is composed of 4 fully-connected layers with a feature size of 40, 30, 20, and 15. We determine the threshold for anomaly score with 5-fold stratified cross-validation with the anomaly data. At the runtime, we detect sensor anomaly a_s for each sensor s from the corresponding HSC model.

5 EVALUATION

5.1 Experiment Setup

5.1.1 Dataset

We evaluate our method on Aruba testbed of CASAS smart home dataset [3]. It includes sensor events of 31 motion sensors and 3 door sensors in a single-resident house. The sensor events include a timestamp, sensor id, motion type (e.g., Sleeping, Meal preparation, Eating), and binary sensor event information (i.e., motion sensor: ON/OFF, door sensor: OPEN/CLOSE). The Aruba dataset only contains normal sensor data. We use sensor events triggered during top-7 frequent motion classes. We randomly choose 20% of the data for testing, and the rest for training.

5.1.2 Baselines

Association Rule Learning. We implement a sensor anomaly detector based on Association Rule Learning (ARL) inspired by prior works [8, 10]. We extract association rules with the apriori algorithm [1] for each motion class and select the rules with a confidence score higher than 0.4. At runtime, a rule r and a segment mj are considered mismatched when all sensors in the antecedent of the rule trigger at least once in the segment, but the consequent sensor is not triggered. We calculate anomaly score $s(x) = \sum_{r \in R_{m_j}} \frac{\text{support}_r}{1.2 - \text{confidence}_r}$ where R_{m_j} is the set of mismatched rules of the segment mj . The sensor whose anomaly score exceeds the pre-defined threshold will be considered an abnormal sensor.

Gaussian Process Regression. We implement an anomaly detection method based on Gaussian Process Regression (GPR) to represent sensor anomaly detection methods based on stochastic methods. FailureSense utilizes Gaussian Mixture Model (GMM) for sensor anomaly detection, but GMM cannot model the relation between multiple sensors, so the anomaly detection accuracy drops significantly. For a fair comparison, instead of using GMM, we apply GPR to enable both sensor-wise anomaly detection and learning multi-sensor relations. GPR is an unsupervised learning method that outputs anomaly probability for the input data. We use the same data pre-processing method as our method. To enable sensor-wise anomaly detection, we train GPR models per each sensor with normal sensor data. We use Radial Basis Function (RBF) kernel to train GPR models. We set the anomaly detection threshold as 0.8 empirically to achieve its best accuracy.

5.1.3 Error Injection

We inject abnormal patterns into the normal data to train our method and evaluate the anomaly detection methods. Based on the sunlight scenario in Section 3.2, we inject errors where the sunlight comes into the

living room, and intermittent sensor fault occurs simultaneously from three sensors (M009, M013, M020) in the living room (Figure 1). To observe the effect of the number of simultaneously faulty sensors, we inject the error to the multiple sensors simultaneously randomly selected among the three sensors.

5.2 Anomaly Detection Accuracy

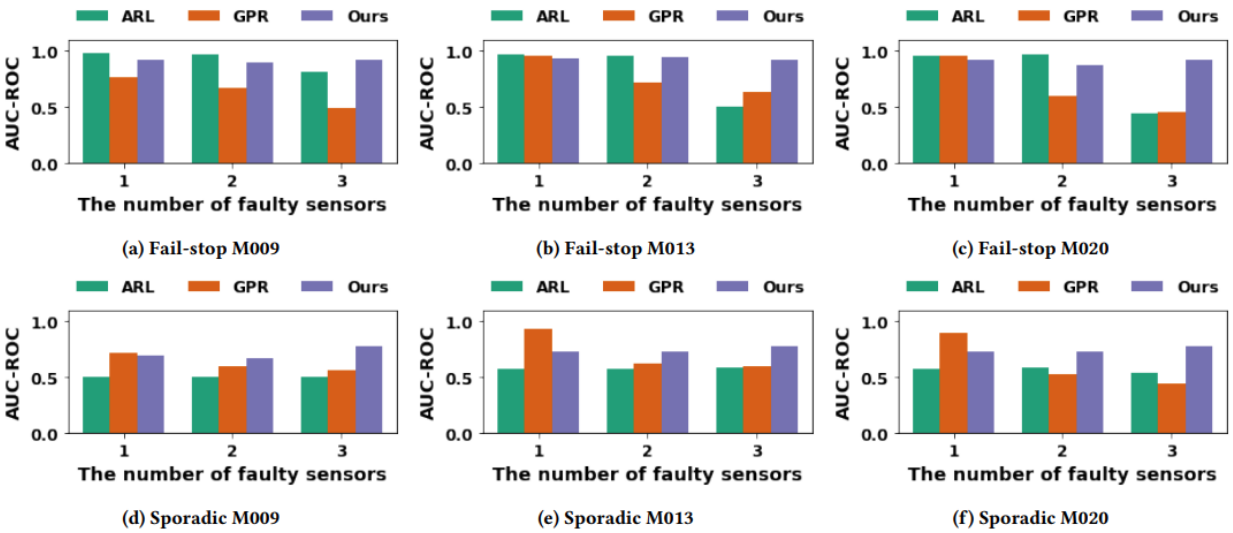

Figure 3: AUC of ROC curves of the anomaly detection accuracy of the baselines and our method for each failure type and detecting target sensors. (a)-(c) show the results for failure-stop failure, and (d)-(f) sporadic failure.

We use two metrics to measure the sensor anomaly detection accuracy: AUC of ROC curves, and F1-score. Figure 3 shows the accuracy of anomaly detection methods in AUC-ROC. Each subfigure shows the results for each failure type and detection target sensor. The number on the x-axis is the total number of anomaly sensors. Our method shows a more stable and robust performance in most settings. In particular, our method outperforms the baselines for all failure types and target sensors when the number of simultaneously faulty sensors is three.

The AUC of ARL (0.960~0.980) is higher than HSC (0.878~0.944) for fail-stop failure when the number of simultaneously faulty sensors is 1 or 2, but the accuracy drops significantly for sporadic failure. This is because ALR has a fundamental limitation in detecting a failure with a stochastic pattern. The AUC of ALR (0.438~0.812) drops significantly when the number of simultaneously faulty sensors is 3, as the assumption of the normal behavior of the non-target sensor does not hold anymore. Our method (0.916 for three sensors) still shows only negligible AUC drops or even increases.

The AUC of GPR (71.6~95.7) is higher than our method (69.5~92.6) only when the number of simultaneously faulty sensors is 1 for both failure types. GPR has some capability to detect anomalies with a stochastic pattern, such as sporadic failure. However, the AUC of GPR (43.5~63.6) drops significantly when the number of the simultaneously faulty sensors is larger than 1, while the AUC of our method shows only negligible drops or even increases. This is because GPR also assumes that the input data x shows normal behavior.

5.3 Threshold Determination

Table 2: F1-scores of the baselines and our method for fail-stop and sporadic failure. The numbers (1, 2, 3) denote the number of simultaneously faulty sensors.

	ARL			GPR			Ours		
#	1	2	3	1	2	3	1	2	3
M009	97.7	96.8	77.2	42.2	34.4	10.3	82.4	89.7	88.7
M013	95.6	96.0	1.7	78.6	16.7	9.7	81.2	87.5	88.7
M020	90.0	97.4	5.1	77.8	22.3	14.5	72.9	84.4	88.7

(a) F1-scores for Fail-stop Failure

	ARL			GPR			Ours		
#	1	2	3	1	2	3	1	2	3
M009	0.0	0.0	2.0	48.4	17.3	12.9	63.5	80.2	74.5
M013	29.7	29.8	32.0	76.1	9.5	2.8	48.0	68.6	74.5
M020	29.7	38.3	35.1	66.9	12.6	7.6	38.7	65.6	74.5

(b) F1-scores for Sporadic Failure

Table 2 shows the result in F1-score. Our method shows high robustness for threshold determination problems. Especially, our method beats the baselines for all failure types and target sensors when the number of simultaneously faulty sensors is three. The F1-score of our method is always higher than that of ARL for sporadic error. The F1-score of ARL is higher than our method when the number of simultaneous fail-stop sensors is smaller than 3, but it drops significantly when the number of faulty sensors increases to 3.

The F1-score of our method is higher or similar to that of GPR for fail-stop failure. The F1-score of GPR (0.669~0.761) is higher than our method (38.7~48.0%) when the number of the simultaneously faulty sensor is 1 for some sensors (M013, M020), but it significantly drops when the number of simultaneously faulty sensors larger than 1 (2.8~17.3%). This is because threshold determination for unsupervised learning methods is fundamentally a tricky problem. Our method explicitly increases the distances between normal and abnormal data on the feature space, making the detection accuracy insensitive to threshold determination (65.6~80.2%).

6 CONCLUSIONS

We develop a new sensor failure detection method for smart homes. The solution addresses not only fail-stop failures on a single sensor but simultaneous sporadic failures on multiple sensors, which can be a common case in practice. We find that existing sensor anomaly detection methods cannot detect these kinds of failures. Our approach is to leverage DNN-based anomaly detection algorithms, which leverage only a small amount of anomalous sensor data to enable sensor-wise anomaly detection. The evaluation results show that our methods beat the baselines regarding anomaly detection accuracy for the simultaneous sporadic failure of multi-sensors.

REFERENCES

[1] 2022. Apriori Algorithm - Wikipedia. Retrieved September 20, 2022 from https://en.wikipedia.org/wiki/Apriori_algorithm

[2] 2022. Association rule learning - Wikipedia. Retrieved September 2, 2022 from https://en.wikipedia.org/wiki/Association_rule_learning

[3] Diane J Cook. 2010. Learning setting-generalized activity models for smart spaces. IEEE intelligent systems 2010, 99 (2010), 1.

[4] Aaron S Crandall and Diane J Cook. 2010. Using a hidden markov model for resident identification. In 2010 Sixth International Conference on Intelligent Environments. IEEE, 74–79.

[5] Kyoung Nam Ha, Kyung Chang Lee, and Suk Lee. 2006. Development of PIR sensor based indoor location detection system for smart home. In 2006 SICE-ICASE International Joint Conference. IEEE, 2162–2167.

[6] Jiawei Han, Jian Pei, and Hanghang Tong. 2022. Data mining: concepts and techniques. Morgan kaufmann.

[7] Timothy W Hnat, Vijay Srinivasan, Jiakang Lu, Tamim I Sookoor, Raymond Dawson, John Stankovic, and Kamin Whitehouse. 2011. The hitchhiker's guide to successful residential sensing deployments. In Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems. 232–245.

[8] Palanivel A Kodeswaran, Ravi Kokku, Sayandeep Sen, and Mudhakar Srivatsa. 2016. Idea: A system for efficient failure management in smart iot environments. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. 43–56.

[9] Madhumita Mallick, Palanivel Kodeswaran, Sayandeep Sen, Ravi Kokku, and Niloy Ganguly. 2018. Tsfs: An integrated approach for event segmentation and adl detection in iot enabled smarthomes. IEEE Transactions on Mobile Computing 18, 11 (2018), 2686–2700.

[10] Madhumita Mallick, Archan Misra, Niloy Ganguly, and Youngki Lee. 2020. DETECTIF: Unified Detection & Correction of IoT Faults in Smart Homes. In 2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM). IEEE, 78–87.

[11] Sirajum Munir and John A Stankovic. 2014. Failuresense: Detecting sensor failure using electrical appliances in the home. In 2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems. IEEE, 73–81.

[12] Nirmalya Roy, Archan Misra, and Diane Cook. 2013. Infrastructure-assisted smartphone-based ADL recognition in multi-inhabitant smart environments. In 2013 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, 38–46.

[13] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In International conference on machine learning. PMLR, 4393–4402.

[14] Lukas Ruff, Robert A Vandermeulen, Billy Joe Franks, Klaus-Robert Müller, and Marius Kloft. 2020. Rethinking assumptions in deep anomaly detection. arXiv preprint arXiv:2006.00339 (2020).

[15] Sasan Saqaeeyan, Hossein Amirkhani, et al. 2020. Anomaly detection in smart homes using bayesian networks. KSII Transactions on Internet and Information Systems (TIIS) 14, 4 (2020), 1796–1816.

[16] Di Wu, Zhongkai Jiang, Xiaofeng Xie, Xuetao Wei, Weiren Yu, and Renfa Li. 2019. LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT. IEEE Transactions on Industrial Informatics 16, 8 (2019), 5244–5253.

[17] Juan Ye, Graeme Stevenson, and Simon Dobson. 2015. Fault detection for binary sensors in smart home environments. In 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, 20–28.