Physics-Based Fault Analysis for Commodity PIR Sensors*

ASHISH KASHINATH, University of Illinois at Urbana-Champaign, Urbana, United States
SIBIN MOHAN, The George Washington University, Washington, United States
AKSHAY NAMBI, Microsoft Research India, Bengaluru, India
SUMUKH MARATHE, Analog Devices, Inc., Boston, United States

Passive Infra-Red (PIR) sensors are ubiquitous and have applications ranging from automatic lighting and heating control in smart buildings, towel dispensers in washrooms, security alarms (for intrusion detection) to human detection robots (for search and rescue). Unfortunately, PIR sensors are prone to failures during deployment due to reasons such as environmental damage, incorrect installation and component degradation among others that can lead to incorrect or faulty data. Currently, such failures are typically detected using either : (a) heavily engineered data-driven, statistical approaches that can have high false positive rates due to unseen data patterns or (b) expensive, unscalable methods that use additional hardware such as video cameras or a golden reference sensor. In this work, we first create a *taxonomy* for the most common PIR sensor failures and analyze these failures from the perspective of sensor physics. We then present PIRMedic—a *physics-driven, edge-based approach* to *detect* and *diagnose* the failures in a PIR sensor using an intrinsic hardware signal *viz.,* the analog output from the pyroelectric element in the sensor. Using this hardware signal in conjunction with frequency analysis and supervised machine learning methods, we obtain a high

*PIRMedic was first introduced as a conference paper in ACM BuildSys [23]. In this work, we expand upon on PIRMedic and provide additional insights on the different classes of faults. We accomplish this by adding a threshold-based technique to study Class I and III faults (Section 4.3) and including experiments using the threshold algorithm to distinguish between them by evaluation in a cafeteria (Section 7.1.4). We also study the impact of incremental failures on Class III faults giving insights into impact of gradual degradation on the A_{out} signal with incremental accumulation of dust(Section 4.2). Additionally, we study the impact of different types of foreign particles such as paper and tape on the sensor performance (Section 4.4). More generally, we include a quantitative study to compare the impact of various faults on sensor performance (Section 3.4 and Table 2). We also study the fault detection accuracy of PIRMedic as a function of pre-deployment training dataset size (Section 7.2). Lastly, we have study the nature of A_{out} from ceiling-mount PIR sensors (Figure 3(c)). We have also made some editorial changes by: (a) adding a goals subsection in Section 1, (b) updating other physics-related sensor failure related work [18, 49], (c) adding a Frequently Asked Questions (FAQs) summarizing some of the common questions in the appendix and lastly (d) including a 3D visualization of the PIR sensor drawn in Tinker CAD and adding more text explaining the different sensor subsystems.

Sumukh Marathe work done while affiliated with Microsoft Research India.
The material in this article is based upon work supported in part by the U.S. National Science Foundation (NSF) under grant NSF CPS 2246937. Any findings, opinions, recommendations, or conclusions expressed in the article are those of the authors and do not necessarily reflect the views of sponsors.
Authors' Contact Information: Ashish Kashinath, University of Illinois at Urbana-Champaign, Urbana, United States; e-mail: ashishk3@illinois.edu; Sibin Mohan, The George Washington University, Washington, United States; e-mail: sibin.mohan@gwu.edu; Akshay Nambi, Microsoft Research India, Bengaluru, Karnataka, India; e-mail: akshay.nambi@microsoft.com; Sumukh Marathe, Analog Devices, Inc., Boston, Massachusetts, United States; e-mail: sumukh.marathe@analog.com.

ACM 1550-4859/2025/07-ART39
https://doi.org/10.1145/3737877

accuracy of 98 – 99% in failure detection and diagnosis. We evaluate our methods using *multiple real-world deployments*, in four distinct locations, in different environment and usage conditions.

CCS Concepts: • **Computer systems organization** → **Reliability**; **Sensors and actuators**; **Embedded systems**; • **Hardware** → **Fault tolerance**; • **Computing methodologies** → **Machine learning**

Additional Key Words and Phrases: Passive infra-red (PIR) sensors, occupancy sensors, fault diagnosis, fault detection, fault isolation, cyber-physical systems, embedded systems, edge computing, edge ML, reliability, dependability, IoT

ACM Reference Format:
Ashish Kashinath, Sibin Mohan, Akshay Nambi, and Sumukh Marathe. 2025. Physics-Based Fault Analysis for Commodity PIR Sensors. *ACM Trans. Sensor Netw.* 21, 4, Article 39 (July 2025), 29 pages. https://doi.org/10.1145/3737877

1 Introduction

Cyber-Physical and IoT Systems such as smart homes, industrial control and healthcare systems often require the correct functioning of a myriad set of sensors. These sensors measure the surrounding environment in the form of either an arithmetic value (e.g., temperature of 72°F) or a logical value (e.g., presence or absence of smoke) and trigger computations and actions. Typically, these sensor data measurements are used by data-driven IoT analytics pipelines (e.g., Microsoft Azure IoT) to derive insights and make higher-level decisions. One of the most versatile and ubiquitous sensors used in numerous modern applications is a **Passive Infra-Red** (**PIR**) sensor. A PIR sensor is a discrete, electronic sensor that captures thermal radiation from human and human-like objects (e.g., animals) to indicate occupancy in a region. They are used in a wide-variety of environments – e.g., in offices to achieve energy savings in buildings (for automatic control of lighting and HVAC systems) and in safety-critical **cyber-physical systems** (**CPS**) like factory floors for intrusion detection. Naturally, these applications mandate the correctness and fidelity of data i.e., the data being a faithful representation of occupancy.

Given the nature of PIR deployments in the wild : for instance in remote, manually inaccessible, harsh conditions (e.g., assembly line of a factory), they are prone to failures and degradation of physical components. These failures could result in false triggers or a failure to capture real movements e.g., assembly line failures in a factory, inaccurate estimation of wildlife counts, conference room lights turning off, or even doors not opening in spite of occupancy. Such sensor failures are often caused by intermittent sensor faults that can happen in any one of its internal physical components [10, 32].

Currently, failures in PIR sensor deployments are usually addressed in one of the following two ways : (a) using an additional, auxiliary sensor such as a CCTV camera combined with image processing algorithms (say to validate the occupancy data) or (b) using high-grade PIR sensors that possesses additional capabilities such as ultrasonic sensors and intricate optics. This makes the sensor deployment very expensive. In addition, deployments need to be refreshed periodically by replacing old sensors with new ones and the former are disposed off adding to both electronic waste and cost. A case in point is our university research building where the occupancy sensor deployed is a high-grade PIR sensor comprising both ultrasonic and infrared sensors, resulting in a *total recurring cost of close to $100000 every 10 years.*[1]

[1]Our 4-floor building (with a total of 244 rooms and 24 aisle areas) has more than 300 occupancy sensors. Each PIR sensor costs $300 [20, 21] with it being refreshed every 10 years, a worst-case recurring cost of $90000 every 10 years, which is expensive.

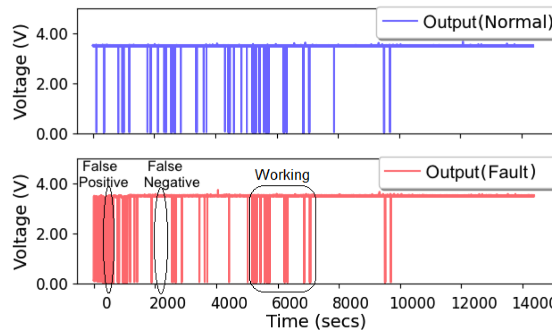


Fig. 1. Sensor data from a working PIR (top) and a faulty PIR (bottom). HIGH indicates no object, HIGH→LOW transition indicates object.

Crucially, *current approaches do not diagnose the cause of failures or perform any analysis of sensor performance*, that can be valuable for IoT engineers during maintenance, repair and testing. For example, in one of our deployments, we observed that failed or degraded sensors can exhibit unpredictable behavior e.g., they can work temporarily for a period of time and then fail sometimes. Figure 1 shows a plot of output data from the sensor vs. time for both a working sensor and a faulty sensor (defective lens) over a period of 4 hours. In this plot, a transition from HIGH→LOW denotes a person coming into the field of view and a value of HIGH indicates that there is no person. We observe that the faulty sensor works intermittently and at other times, can miss a person (false negative) or incorrectly flag the presence of a person (false positive). Missing a person can cause a potentially critical failure in applications such as door opening systems or emergency shutdowns on factory floors. We show that accurate detection of such failures is possible by *characterizing the internals of the sensor.*

Our solution, PIRMedic detects and diagnoses failures of PIR sensors by characterizing the physics of sensing viz., the fresnel lens optics and the pyroelectric effect. To keep the cost of the deployment low, PIRMedic is implemented on the edge platforms of the deployment and targets commodity PIR sensors. In this process, we *devised a taxonomy of key failures for aiding diagnosis and repair.*

Specifically, we demonstrate that a signal intrinsic to PIR sensors *viz.,* the *intermediate analog output from the pyroelectric element, referred to as A_{out}* (Figure 3(a)) can be used for fault detection and diagnosis. We utilize this signal to derive information about reliability of the PIR sensor platform as well as to perform an analysis into the different failure modalities. *To the best of our knowledge, this is the first edge-based, low-cost approach to use an intrinsic signal (A_{out}) for fault detection and diagnosis of PIR sensors.* We analyze the behavior of A_{out} in detail, under practical deployments, with realistic occupancy over a wide variety of observed failures.

Summary of Contributions:

(1) *From physics to failures:* We use the working of a PIR sensor to understand the process of object detection and infer the key points where failures could lead to incorrect sensor data,
(2) *Failure taxonomy:* We study the impact of different failures on the sensor performance and systematize the key failures into a failure taxonomy. We also provide a quantitative method to study the severity of failures.
(3) *Non-intrusive, online fault detection and diagnosis:* We present an online technique (does not require any disassembling), *implemented at the edge*, for failure detection as well as diagnosis by utilizing the intermediate, analog output from the pyroelectric element in the PIR sensor (A_{out}).
(4) *Insights from Real-world Deployments:* We show the efficacy of the proposed techniques using a deployment of 15 PIR sensors in four practical occupancy scenarios over a period of 3 months.

Goals:

(1) *Detect failures:* Sensors can give faulty data due to the incorrect operation of various physical components involved in the sensing process. We seek to detect and isolate such failed sensors.
(2) *Diagnose failures:* In addition to detecting failures, we perform failure diagnosis to identify the cause of the failure. We use domain-knowledge of the sensing physics and our understanding of how different failures can impact the physics. This aids in repair and performing quality-control checks.
(3) *Edge-based solution:* PIRMedic is tailored for edge platforms (Raspberry Pi, Arduino Microcontrollers) that are interfaced directly to multiple sensors, say via GPIO. Edge-based reliability solutions allow us to reject data from faulty sensors in IoT deployments thus saving the cost of both (a) the edge to cloud network bandwidth and (b) cloud compute consumed by faulty data.
(4) *Target Audience:* PIRMedic benefits two sections of the community: (a) engineers developing and deploying IoT/Edge computing for occupancy sensors (e.g., Smart Buildings), (b) manufacturers of occupancy sensors can use the insights from this work toward future PIR sensor hardware designs.

2 Related Work

Current solutions for failure detection and diagnosis in sensors fall into one of three categories—(a) Data-driven techniques, (b) Calibration-based techniques, and (c) Fingerprint-based techniques.

Data-driven techniques. Prevalent research efforts have largely focused on data-centric approaches (rule-based or anomaly detection), where historical data of the sensor is analyzed and a fault is identified if the data is out of bounds of the expected behavior [43, 44]. Sharma et al. [47] proposed a multiplicative seasonal ARIMA time series model for fault detection, where the parameter captures periodic behavior in the sensor data. The downside of temporal analysis methods is that they are prone to false positives and are not feasible in long-term deployments. Wu et al. [56] used a spatial mining-based approach that uses spatial correlation between neighboring sensors to detect anomalies. Additionally, techniques such as Ayadi et al. [6], Murphree [36], and Power et al. [43] require significant labeled data as it models only the environment and does not model the sensor physics. Estimation-based methods model normal sensor behavior leveraging spatiotemporal correlation and probabilistic models such as Bayesian or Gaussian distributions [24] and they work well in homogeneous environments. In a nutshell, the key challenge with data-driven techniques is that it relies on rules and patterns in the sensed data. We argue that for PIR sensors since the data is non-periodic and dependent on deployment scenarios (e.g., people counting in Starbucks, animal detection in forests), *it is non-trivial to detect faults by just analyzing sensor data.* Further, this requires significant manual efforts and tailor-made rules to detect faults, and can eventually have high false positives due to unseen data patterns. Additionally, data-driven techniques do not provide information about the specific nature or degree of failure in the sensor. As a result, they do not aid in the process of developing repair strategies. This is also seen in works such as Tawakuli et al. [52], that discusses the repercussions of missing data to be of two types—(a) isolated missing data and (b) sequence missing data. They develop a domain-independent, dataset-independent, hybrid imputation strategy that can give better estimation accuracy by looking at six real-world datasets such as AirQuality and NASDAQ time-series data. Tawakuli et al. [51] describe a generic edge-cloud IoT-based preprocessing framework targeting AI pipelines. The framework comprises steps such as data cleaning and feature engineering, and can leverage PIRMedic to assist in cleaning data associated with faulty PIR sensors. In summary, data-driven techniques, while scalable, often entail substantial labeling efforts, leading to high costs, and can exhibit low accuracy due to their dependence on custom rules and patterns. Moreover, they provide no guidance on correcting faulty sensors, as they disregard the underlying physics of sensing.

Calibration-based techniques. These techniques rely on the presence of an additional (reference) sensor, either to perform periodic calibration [31, 55] or carrying additional information in a different domain space. For example, using a camera to cross-check the data of a PIR sensor [50]. Numerous algorithms have been developed for performing calibration such as blind calibration [16, 53], collaborative calibration [46, 57] and transfer calibration [58]. Agarwal et al. [1] use temperature and power consumption data to infer if a HVAC is faulty or working. It relies on the correlation between the two individual data streams from auxiliary sensors *viz.,* smart meters and temperature sensor. By studying such a relationship between multiple sensors, Agarwal et al. [2] provide a optimization framework to minimize the number of sensors required to sense a facet such as occupancy. Their framework uses fault tolerance information in their framework and, in such a setting, PIRMedic can be used to minimize the number of sensors in a **Building Management System** (**BMS**), thereby lowering operational costs and improving efficiency. Huchuk et al. [22] develop a machine learning model based on data from thermostats to predict set points overrides that happen across a set of buildings, helping create more energy efficiency schemes. They note that using building-level indoor conditions will help create more energy efficient schedules and PIRMedic can aid in improving the accuracy of such systems. However, all of these approaches that use reference sensors are expensive and lacks scalability for large IoT deployments.

Fingerprint-based techniques. Chakraborty et al. [10] develop a sensor signature *viz.,* "Fall Curve" that measures a sensor's voltage response when the power is turned off, to detect faults in periodic on-off based analog sensors. The Fall Curve characterizes the parasitic capacitance and hardware circuitry of the sensor. Crucially, Chakraborty et al. show this signature is unique for every sensor type and is independent of the environment. They demonstrate that this approach works well for sensors that have very little or no warm-up time and work in a periodic on-off fashion. Furthermore, Chakraborty et al. [10] point out to the fact that extending this approach to digital sensors where one doesn't have direct access to the analog signal is challenging. Additionally, Fall Curve is complex in sensors having low value of operating voltages and lack periodicity in the sensing mechanism. Our work concerns this design space of exploring faults in discrete-valued PIR sensors. PIR sensors are challenging since in addition to being discrete valued, the internal analog signal *viz.,* output from the pyroelectric element has a very low value. Similarly, Tambe et al. [49] show a variant of Fall Curve i.e., "Fall Time" of the analog signal can detect faults and drifts in phototransistor components of the sensor. However, both Fall Curve and Fall Time signatures do not work on certain types of sensors including PIR, that operate under low voltages and are event-triggered. Marathe et al. [32] show that analyzing the current profiles of a digital sensor can provide insights into component failures especially in electro-mechanical sensors. However, this approach is limited to power-hungry digital sensors. Gupta et al. [18] develop a **Verified Telemetry SDK** (**VT-SDK**) constructed on robust sensor fingerprinting algorithms and demonstrate fault detection on multiple sensors ranging from air pollution to pressure sensors. There are other hardware-based techniques such as Liu et al. [28] that detects electronic faults by developing additional circuits inspired by the biology of neurons. In summary, fingerprint-based techniques have been applied to sensors such as air pollution, smoke, and soil temperature and moisture sensors but have not been extended to PIR sensors.

PIR sensors are challenging since in addition to being discrete-valued, the internal analog signal output from the pyroelectric sensor has a very low value, often oscillating between 1–1.8V. We show that the intermediate analog output (A_{out}) from the PIR sensor carries interesting insights on the characteristics of the sensor. Narayana et al. [37] leveraged this signal by developing a customized sensor array for performing localization and studying object metrics. They develop a customized sensor tower consisting of a strategic geometric arrangement of multiple PIR sensors to enable capturing the analog signal from those multiple sensors. They use this to study the object size (height and width) and posture (such as its distance from the sensor and the direction of motion) and its impact on features of the analog signal such as frequency, amplitude, phase, and time difference between peaks. However, Narayana et al. [37] do not explore reliability.

3 PIR Sensors

We take a deep dive into a PIR sensor with three goals—(a) understanding the physics of sensing (Sections 3.1, 3.2), (b) correlating physics to the failure scenarios (Section 3.3), and (c) analyzing the effect of these failures on the sensing performance (Section 3.4).

3.1 Background

The teardown of a PIR sensor, showing the internal components is shown in Figure 2. It comprises three subsystems—(a) a lens, (b) a pyroelectric attached to an optical filter, and (c) an electronic subsystem. These subsystems act in sequence i.e., Lens → Pyroelectric → Electronic to perform the end-to-end sensing process as shown in Figure 3(a). We describe the different subsystems next.

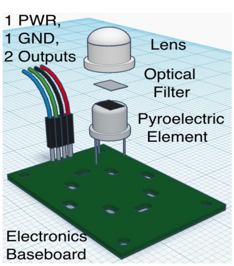


Fig. 2. PIR sensor teardown showing the different subsystems illustrated in TinkerCAD.

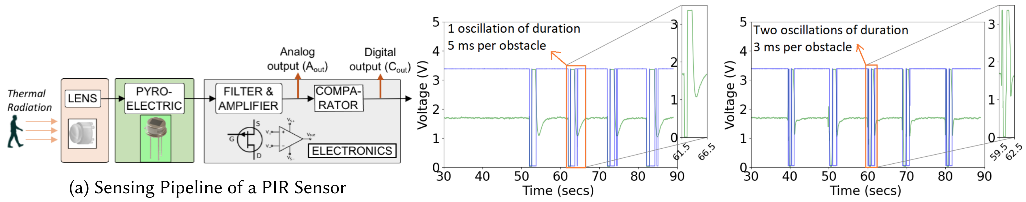
Lens Subsystem A plastic fresnel lens [5] is used to capture and precisely focus thermal radiation into an optical filter that is aimed at a pyroelectric element. Fresnel lenses have a large capture area (aperture) and are used to concentrate radiation into a narrow beam. They are used to divide the entire region observable to the PIR sensor into small cells, which increases range of the sensor by manifolds. They have a large capture area (aperture) and are common in a wide variety of applications such as lighthouses, solar cells, and even in optical landing systems in aircrafts [54].

Pyroelectric Subsystem comprises of an optical filter and a pair of pyroelectric elements. The optical filter is designed to filter out thermal radiations from wavelengths outside the human range (i.e., 5 μm to 10 μm), which is then incident on the pyroelectric elements. The pyroelectric elements convert the thermal radiation into an electrical voltage signal, a process known as the *pyroelectric effect*. The output of the pyroelectric subsystem is an analog signal that is sent to the electronic subsystem. In practice, there are two pyroelectric elements connected in series in a differential configuration. This differential configuration enables the PIR sensor to detect *movement* of objects as static objects will trigger both the pyroelectric elements equally and cancel out unlike moving objects that trigger both elements unequally resulting in an output from the pyroelectric subsystem. Due to the differential output voltage, the pyroelectric subsystem does not measure the absolute intensity of incident thermal radiation but measures a change in incident thermal radiation, say caused by motion. Note that the pyroelectric element by itself is passive i.e., has no power source.

Electronic Subsystem consists of a filter (RC filter), amplifier (JFET or OpAmp), and comparator (OpAmp) circuits. The output from pyroelectric subsystem are filtered to remove noise, amplified to increase its magnitude, and finally sent to a comparator to convert analog signals to discrete signals. This *discrete signal is HIGH when there is no motion and goes HIGH→LOW when motion is detected*. Systems connected to a PIR sensor such as lighting or heating use the discrete signal to regulate the illumination or temperature in a building.

3.2 Analysis of Output Signals of a PIR Sensor

There are two output signals from a PIR sensor as shown in Figure 3(a):



(a) Sensing Pipeline of a PIR Sensor (b) Inline lens (wall-mount) (c) Round lens (ceiling-mount)

Fig. 3. Effect of different types of lens on the A_{out} curve, shown in green—the inline lens type produces a single oscillation whereas a round lens type produces two oscillation per obstacle. The corresponding C_{out}, shown in blue, stays HIGH when there is no motion and goes LOW when motion is detected.

(1) **Final discrete output** of the electronic subsystem (C_{out}) that follows negative logic i.e., stays HIGH when there is no motion and goes LOW when motion is detected.
(2) **Intermediate analog output** just prior to the discretization process (A_{out}).

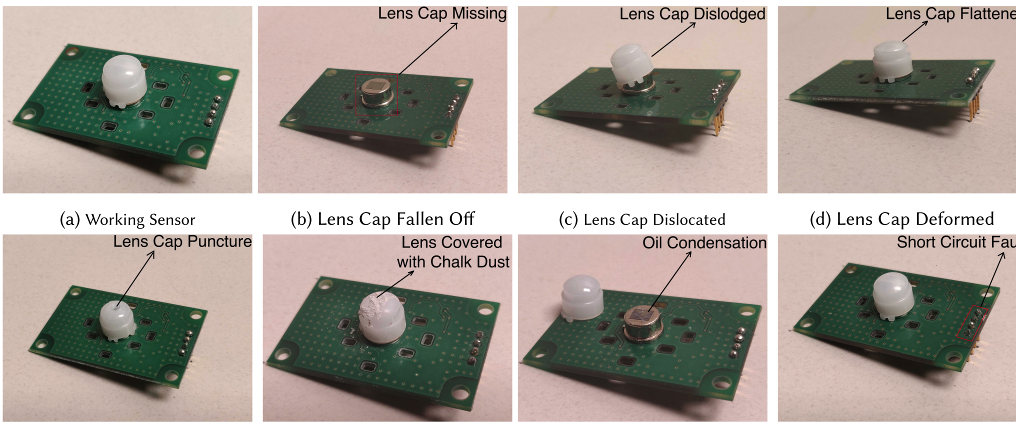
To analyze C_{out} and A_{out} during operation, we interface a PIR sensor to an Arduino microcontroller [48]. Figure 3(b)–(c) shows plots of A_{out} and C_{out} for a period of 60 seconds, both in the presence and absence of an obstacle. The y-axis represents voltages seen at the output and x-axis represents time. As expected, C_{out}, marked in blue, goes LOW when an obstacle comes into the field of view and stays HIGH otherwise. On the other hand, A_{out}, marked in green, measures around 1.8 V when there is no obstacle and produces oscillation(s) when an obstacle comes into the field of view (shown magnified). The nature and shape of the A_{out} curve also depends on the nature of lens used in the PIR sensor. Typically, an inline lens is used for the wall-mount version and a round lens is used for the ceiling-mount version. Figure 3 also shows the variations in A_{out} between the two lens types—(a) Inline lens has a wider curve compared to round lens and (b) Round lens has double number of oscillations of A_{out} curve per obstacle. We used a wall-mount PIR sensor in our experiments though our analysis, techniques, and observations are equally applicable to ceiling-mount sensors. With a working background on PIR sensors, we next study the various failures in a PIR sensor and its impact on sensing physics.

3.3 Failures in PIR Sensors—A Taxonomy

Table 1. **Taxonomy of Failures** in a PIR Sensor

Failure	Definition & Impact
Lens Subsystem	
Lens dislodged(Class I)	Lens cap suffers partial or complete dislocation e.g., physical impact with a foreign object, degradation of bonding, etc.
Lens deformed(Class II)	Lens cap suffers physical damage in-place e.g., deformation, puncture, etc.
Lens covered(Class III)	Lens cap gets physically obstructed by foreign particles e.g., dust, paper or tape
Pyroelectric Subsystem	
Optical filterdamage (Class IV)	Damaged by environmental factors e.g., oil condensation
Electronic Subsystem	
Electronic faults(Class V)	Hardware failures e.g., short circuits, floating outputs etc.

Fundamentally, failures in a PIR sensor could occur either on the lens, the pyroelectric element or the electronics. We *analyze and categorize practical, common and most frequent* failures as gathered in discussions with IoT engineers and technicians who install and maintain PIR sensor deployments. We developed a new taxonomy for such failures, as summarized in Table 1 and visualized in Figure 4.



(a) Working Sensor (b) Lens Cap Fallen Off (c) Lens Cap Dislocated (d) Lens Cap Deformed

(e) Lens Cap Punctured (f) Lens Cap Covered (g) Optical Filter Damaged (h) Electronic Fault

Fig. 4. Some sensors used in our study. Note that the failures are not always visually perceivable as the sensors are typically inaccessible.

3.3.1 Failures in the Lens Subsystem. The lens is geometrically constructed to precisely focus the thermal rays onto the pyroelectric element. As a result, failures affecting the optical integrity of the lens can result in loss of precision for focusing the thermal rays. We observe three types of failures here, termed Class I–III failures.

<u>Lens dislocation (Class I)</u>: As the lens is stuck on the sensor board with an adhesive or machined to fit over the pyroelectric element, it could get dislodged from its place or completely fall off the sensor board due to factors such as thermal expansion or physical impact.

<u>Lens deformation (Class II)</u>: As the lens is made of flexible, 0.4mm thick high density plastic [38], it is susceptible to deformation by physical damage. This can alter the curvature or puncture the lens. As in Class I, the consequence of failure is imperfect capture and focusing of the thermal radiation incident on the PIR sensor.

<u>Lens hindrance (Class III)</u>: Dust particles (e.g., factory floors, building construction) or common objects such as paper and plastic tape can absorb the thermal radiation resulting in reduced to no radiation falling on the pyroelectric element. The deposition of particulate matter also causes dispersion.

In summary, Classes I–III failures affect the core functionality of the lens, and results in the imperfect capture and focus of thermal radiation.

3.3.2 Failures in the Pyroelectric Subsystem. Optical Filter Damage (Class IV) The optical filter and pyroelectric elements are prone to degradation with exposure to high temperature and humidity. As the optical filter is carefully calibrated to trigger in the region of human motion, it is susceptible to damage due to environmental factors such as heat sources (e.g., room heaters, air-conditioners) and humidity (e.g., oil condensation) as it affects the perceived temperature of the object. We name these as Class IV failures.

3.3.3 Failures in the Electronics. On-board electronics consisting of filter, amplifier, and comparator circuitry are prone to failures such as shorted or floating pins that we refer to as Class V failures. We only consider simple, electronic failures—such as open, short circuits and floating inputs. Vendor-based IC testing procedures and board design process advancements have made electronic failures such as regulator, bus failures and PCB malfunctions infrequent in practice.

Table 2. **Misclassifications** Analysis for Different Failures

Failure	Lens Dislodged	Lens Covered	Heat Damage	Humidity Damage
${\delta_N}^1$	35	48	29	48
${\delta_N}^2$	69	94	53	87

The numbers indicate the deviation (as defined in Equation (1) between C_{out} values) of a faulty sensor from that of a working sensor. Higher number indicates increased degree of failure.
[1] 6 hour run, [2] 12 hour run.

Summary: Failures in PIR sensors can occur in either the lens subsystem, the pyroelectric subsystem, or the on-board electronics, each of which manifest differently in the underlying physics.

3.4 Analyzing Severity of Failures—A Quantitative Analysis

A failure is said to impact the sensor performance if the failure results in either **false detections** (**FD**) i.e., false positives or **missed obstacles** (**MO**) i.e., false negatives in the sensor output (C_{out}), compared to a working sensor. As seen in Figure 3, C_{out} is HIGH (1) when no obstacle is detected and C_{out} goes HIGH (1) → LOW (0) when an obstacle is detected.

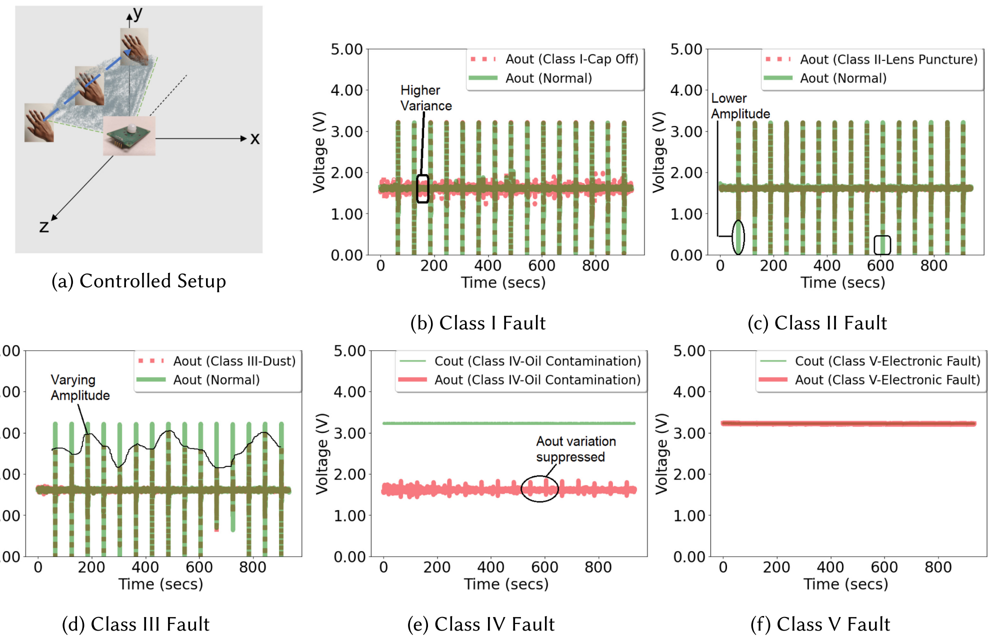
Controlled Failure Study: To understand the impact on sensor performance, in our lab, we take two PIR sensors, one which is without the fault (S_{working}) and one with the fault (S_{tampered}). The two sensors are placed in as close a proximity to one another as possible such that—(a) distance between the sensors are closer than the size of obstacle, (b) the obstacle is moved in a plane such that it comes into the detection region of both the PIR sensors simultaneously. This implies that when the obstacle (we used our palm in this case) is moved across the detection region of both PIR sensors, it grazed both at the same time, triggering the same responses from both the sensors.

We then calculate the absolute difference (δ_N) between the C_{out} values of the tampered sensor *viz.*, $C_{\text{out,tampered}}$ and the reference working sensor *viz.*, $C_{\text{out,working}}$ by summing over all observations (Refer Equation (1)). Intuitively, a relatively low δ_N would imply that the tampered sensor is still in acceptable condition i.e., the failure is not having much impact on the operation of the sensor. On the other hand, a high value of difference implies that the failure is much serious. To ease computation, we can also compute δ_N by dividing the time series into equal size chunks called *windows* (W) and computing the difference between average C_{out} values of S_{tampered} and S_{working} in that window.

Table 2 summarizes δ_N for a few types of failures as observed over 6-hour and 12-hour runs deployed in the elevator of our building. The numbers in the column represent deviation or misclassifications (δ_N) i.e., the number of FD (both false positives and false negatives) by the faulty sensor calculated using Equation (1). We used a window size, W = 1024 samples, which comes to a window of time duration slightly over 51 seconds at a sampling rate of 20 Hz. In our case, the lens covered failure (8 failures per hour) is more pernicious than lens having fallen off (6 failures per hour).

$$\underbrace{\boxed{\delta_N}}_{\text{misclassifications}} = \sum_{windows} \left| \overbrace{\boxed{C_{\text{out,tampered}}}}^{\text{tampered}} - \overbrace{\boxed{C_{\text{out, working}}}}^{\text{working}} \right|, \tag{1}$$

Summary: Quantifying the impact of a failure gives hints on how pernicious specific failures are *once they have been detected.*



(a) Controlled Setup (b) Class I Fault (c) Class II Fault

(d) Class III Fault (e) Class IV Fault (f) Class V Fault

Fig. 5. ((b)–(f)) A_{out} waveforms for sensor faults of Classes I–V. Note that in Class I – III faults, despite the obstacle still being detected, A_{out} provides indication of underlying abnormalities in the sensor.

4 Characterizing Failures Using an Intrinsic Signal

Our hypothesis is that the intermediate analog signal (A_{out}) from the pyroelectric element captures information that is critical to detect various failures. We now describe how this intrinsic signal and its underlying physics is useful to characterize failures.

4.1 A_{out} Signal to Detect Failures in PIR Sensors

As described in § 3.2, there are two output signals from a PIR sensor, *viz.*, A_{out} and C_{out}. The latter is a discrete signal typically used for detecting obstacle presence and is derived from A_{out}. To show the utility of A_{out}, we conducted numerous controlled experiments where we manually injected commonly seen faults (Figure 4).

Using the setup from Section 3.4, we co-located two sensors, S_{tampered} and S_{working}, where we change the S_{tampered} sensor to test different types of failures described in Section 3.3. While in Section 3.4, we focused on C_{out} to study sensor performance during failure, in this experiment we additionally analyze A_{out} to get an understanding of the *sensor physics* during failure.

We note the output signals (i.e., A_{out} and C_{out}) from S_{tampered} in every failure scenario and compare it with that from S_{working} *to understand the impact of failure on physics of the sensors.* Each experiment described below lasted for 15 minutes, where the obstacle (our palm in this case) was moved into the region of detection once every minute, as shown in Figure 5(a). We plot A_{out} for both a working sensor and every type of faulty sensor in Figure 5(b)–(f). The y-axis represents voltage output of A_{out} and x-axis represents time. For the working sensor S_{working}, we expect a spike in A_{out} once every 60 seconds denoting motion of the palm. Let us now study the nature of A_{out} in each fault.

Lens dislocation (Class I) faults. The absence/dislocation of the lens causes imperfect focus of thermal radiation resulting in some residual output at the pyroelectric element even when there is no obstacle present. The impact of a Class I fault on A_{out} is shown in Figure 5(b). We can see that both the tampered and working sensors are still able to detect the obstacle (as shown by the 15 vertical spikes of A_{out}). However, when the obstacle is not present, we observe that the noise of A_{out} is much higher with a Class I fault as compared to a working sensor. Thus, it is important to look beyond just C_{out} signal to identify such failures.

Lens deformation (Class II) faults. Class II faults can potentially lead to missing obstacles at the periphery of the field of view due to deformation or loss of material integrity (e.g., puncture). Figure 5(c) shows the A_{out} for a sensor with a Class II fault. The amplitude of A_{out} at the times of obstacle detection is lower when compared to a working sensor. This is expected since the damage to lens leads to reduced thermal radiation incident on the pyroelectric subsystem. In general, Class II faults create blind spots in the lens aperture that can reduce amplitude of the A_{out} signal.

Lens hindrance (Class III) faults are caused due to foreign particles entering the lens that compromise its optical capability, thereby affecting the intensity and angle of radiation incident on the pyroelectric element. We induce such faults by depositing some dust on the lens. Figure 5(d) shows that the amplitude of A_{out} varies at each of the detection points. The variation depended on the amount of dust with respect to the orientation of the obstacle. Though the obstacle was still being detected, we note that with increased dust deposition, the amplitude of A_{out} can fall below the comparator threshold required to capture the obstacle, resulting in a MO i.e., false negative.

Failure in pyroelectric components (Class IV). Class IV faults happen when the optical filter on the pyroelectric element comes in contact with contaminants such as oil, mist, or other smudge, resulting in potentially missing obstacles. High temperatures can also cause expansion of the optical filter that can also result in failures. We induce Class IV faults by spraying some oil on the optical filter. We observe in Figure 5(e) that A_{out} has significantly attenuated spikes at each of the points that correspond to obstacle motion, but the spikes are not high enough to cause C_{out} to drop LOW. This results in the obstacle to be missed completely. A similar effect is observed when the optical filter is exposed to heat as it affects the temperature variation needed for pyroelectric effect.

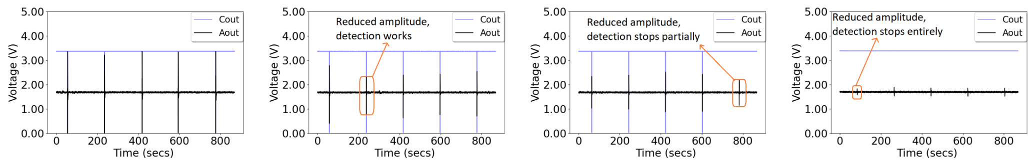
Failure in electronics (Class V) faults. Class V faults are typically electronic faults such as short or open circuits. They can cause the A_{out} or C_{out} values to be "stuck" at certain anomalous value such as HIGH (3.3 V) or LOW (0 V). As a result, these failures can cause the obstacle to be completely missed. This is observed in Figure 5(f), where A_{out} is shorted to the power supply, resulting in C_{out} following A_{out} and remaining HIGH. Consequently, the A_{out} waveform here is a flat, horizontal line and lacks oscillations.

Summary: The core insight here is that A_{out} can shed light on the physical and electrical operating conditions of a PIR sensor as opposed to the mere boolean occupancy indicated by C_{out}. Thus, being able to snoop in on A_{out} can capture the interaction of IR radiation on the different sensor subsystems.

4.2 Analysis of A_{out} During Partial Failures and Gradual Degradation

In Section 4.1, we observed the impact of failures in PIR sensors, from classes I through V on nature of A_{out} and C_{out}. While we developed a structural taxonomy in Section 3.3, we note that failures need not be binary in nature, and there can be partial failures. Consequently, there are variations in each of the failure classes. For example, in Class III faults, the gradual build of dust can lead to degradation of sensing performance.

To study the effect of gradual failures occurring in Class III, we divided a spatula of chalk-dust (approximately one-fourth of a teaspoon) and measured the effect of dust on response of the sensor. At every stage of study, we incrementally added dust and measured performance by plotting the



(a) Clean Sensor : Working (b) Degraded Sensor : Still Working (c) Degraded Sensor : 1/5 miss (d) Failed Sensor : 5/5 miss

Fig. 6. Dust gradually building up in different degrees causing Class III faults – (a) is a clean sensor and dust gradually builds up (b) – (d) at which stops detecting obstacles.

two output signals C_{out} and A_{out} in the presence of an obstacle. The experiment is summarized in Figure 6. With increasing amount of dust being deposited, we see that the final discrete output misses the obstacle on occasion while detecting it sometimes. As a result, information about the dust deposition is lost at times from the final discrete output, C_{out}. *Note that however, the intermediate analog output, A_{out} is able to capture information regarding the deposition of dust and by showing the degradation of sensing performance via its reduced amplitude.* Consequently, the obstacle detection process works till a reduced amplitude of A_{out} (Figure 6(a)–(b)), below which the sensor begins missing some obstacle instances (Figure 6(c)). This process aggravates as more dust is accumulated on the sensor until a stage when there is a complete failure (Figure 6(d)).

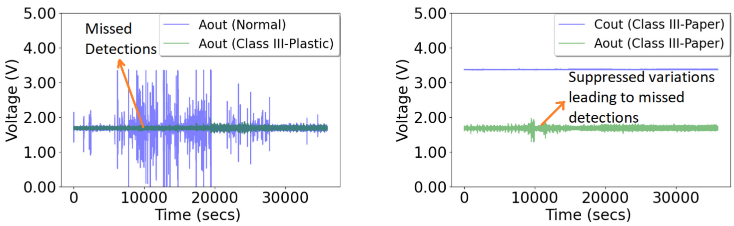
Summary: The key observation here is that capturing the physics of sensing via A_{out} helps glean valuable insights into even partial failures, which is completely absent in the C_{out} signal.

4.3 Analysis of A_{out} During Known Quiet Times

We draw some intuition on the behavior of A_{out} during known times of no occupancy, e.g., post business hours in case of office buildings. The intuition stems from the arrangement of pyroelectric elements and its interplay with the lens subsystem. In a working sensor, we observe that due to the precise geometric arrangement of lens and pyroelectric elements, when there is no obstacle, the thermal radiation from the environment falls equally on the two pyroelectric elements, causing it to cancel out and leave very little noise at the output of pyroelectric subsystem. However, when the lens is covered by a foreign substance (i.e., Class III fault), some of the thermal radiation is absorbed by the substance, before falling onto the pyroelectric elements. This results in a much lower noise at the output of pyroelectric subsystem. Alternately, if the lens is dislodged (i.e., Class I fault), the two pyroelectric elements are not perfectly matched in differential configuration, which leads to higher noise at output of the pyroelectric subsystem. We observe this noise by measuring the *noise floor* in the variance of A_{out} signal, during times of no obstacle.

Thus, we note that variance of A_{out} during a period with no occupancy can be used to get hints on Class I (lens dislodged) and Class III faults (lens covered). In particular, compared to a working sensor where the lens is mounted correctly, we expect a drop in variance when the lens is covered and a rise in variance when the lens is dislodged. In other words, $\sigma(A_{\text{out}})|_{Class\ III} \leq \sigma(A_{\text{out}})|_{Working} \leq \sigma(A_{\text{out}})|_{Class\ I}$, where $\sigma(A_{\text{out}})|_X$ denotes the variance of A_{out} under scenario X. We can use this idea to develop a preliminary fault detection algorithm that can be used to screen for lens faults (Class I and III). This requires computing two variance-based thresholds (T_L, T_H) to deduce status of the lens. However, we note that using the variance of A_{out} is not viable in all practical conditions. In particular, using this method works only during *known quiet* times i.e., in the known absence of occupancy which can be a *disadvantage* for some deployments. As a result, this approach is suited for preliminary sensor health checks outside office or regular business hours of operation such as weekends.

4.4 Analysis of Different Types of Foreign Particles



(a) Plastic (b) Paper

Fig. 7. Effect of different foreign particles producing Class III faults of different degree – the figures show that when the lens is obstructed with foreign particles like plastic tape and paper, it could result in complete failure.

Different materials have distinct thermal characteristics and thus affect the A_{out} response differently in case of Class III faults. Stated alternately, the amplitude shaping of A_{out} varies depends on the material and its quantity. There are three kinds of foreign particles we study in our setup—(a) paper, (b) plastic tape, and (c) dust. We saw in Figure 5(d) how deposition of dust can compromise with the A_{out} responses. Figure 7 shows Class III fault scenarios repeated with plastic tape and paper. In each of these runs, a normal sensor (represented in the blue curve in Figure 7(a)) detects obstacles as demonstrated by the A_{out} variation, whereas foreign particles such as plastic tape (Figure 7(a)) and paper (Figure 7(b)) can lead to a *complete failure* of the PIR sensor.

4.5 Frequency Domain Characterization

We have showed A_{out} signal has hints about various failure scenarios. To get additional insights, we transform the time domain A_{out} signal into frequency domain using the **Fast Fourier transform** (**FFT**) to derive robust features. FFT offers high resolution in the frequency domain and deconstructs the frequencies and harmonics present in a signal. Frequency domain representations have been used to characterize human activities using smartphone sensors and perform vibration screening for faults both in aircraft and railway lines [4, 19].

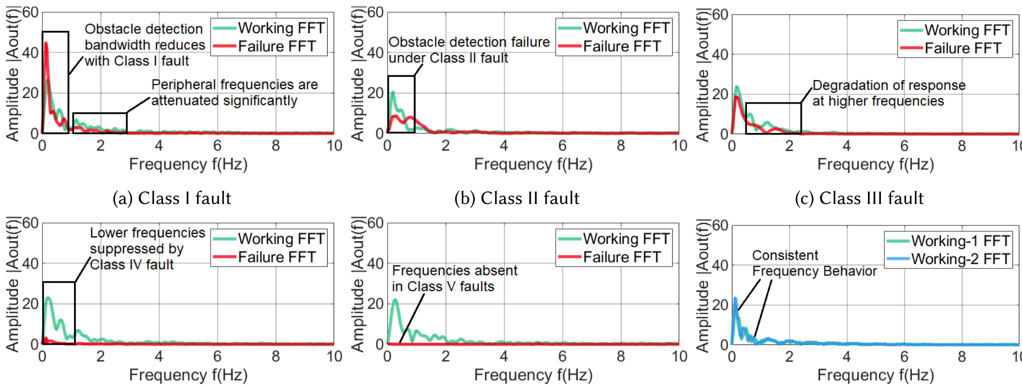
We plot the FFT representations of A_{out} corresponding to different faults in Figure 8(a)–(e) as compared to a working sensor, all under the presence of an obstacle. The y-axis plots magnitude of FFT coefficients and the x-axis plots frequencies in the human range of motion (0 – 10 Hz). In case of a working sensor, we observe two big peaks and additional peripheral frequencies up to 4 Hz. We now study how the frequency spectrum varies depending on the type of faults.

Class I faults We observe in Figure 8(a) that a dislodged lens cap (either partial or complete) leads to reduced information capture (lower bandwidth) and as a result, the sensitivity in the periphery of the sensors reduces. Note that there is just a single prominent peak in the faulty sensor (at a slightly lower frequency when compared to the normal sensor) that corresponds to the obstacle being perfectly aligned with the center of sensor as it passes. The magnitude reduces sharply as it moves away from the center of the sensor.

Class II faults Figure 8(b) plots the FFT for a sensor with deformed lens cap. We note that Class II faults lead to significantly suppressed primary peaks (approx. 10 db). In addition, the peripheral frequencies are attenuated as observed in Class I. These faults can manifest itself in reduced sensor performance as some of the detection region become *blind spots* for the sensor.

Class III faults due to accumulation of foreign particles in the lens, lead to reduced sharpness of frequencies. We observe in Figure 8(c) that the degradation starts with damping of the higher frequencies. As the amount of dust increases, we observe the frequencies getting increasingly damped until the object detection starts failing.

Class IV faults We observe in Figure 8(d) that the frequencies present in the output are heavily suppressed resulting in MO detection. This is expected since the optical filter plays a major role



(a) Class I fault (b) Class II fault (c) Class III fault

(d) Class IV fault (e) Class V fault (f) **Comparing** A_{out} of two working sensors

Fig. 8. Frequency domain representations of A_{out} under different faults in a PIR sensor. Only frequencies upto 10 Hz are shown as most PIR sensors detect human motion within 10 Hz.

in filtering out the non-human range frequencies from the thermal radiation and only passing through the frequencies corresponding to the human range. The presence of oil condensation or heat sinks causes absorption of the thermal radiation resulting in insufficient heat to cause pyroelectric effect.

Class V faults such as open and short circuits are easy to detect due to the absence of any prominent frequencies. Consequently, A_{out} is *stuck at* a value leading to only dc frequencies being present in the output. This result is observed in Figure 8(e), where the C_{out} and A_{out} pins are shorted on the board.

Summary: Frequencies present in FFT representations of A_{out} provide hints regarding status (i.e., working/failed) of PIR sensors.

5 Fault Detection and Diagnosis

In Section 4, we analyzed the impact of failures on the characteristics of the A_{out} output signal in both time domain and in frequency domain using FFT. We now investigate if the characteristics of A_{out} contain behavior that is amenable to learning i.e., if there are features to be learnt. First, we analyze the learnability of the fault detection and diagnosis problem in Section 5.1. During this, we show that in every type of failure, there exists information about the failure and quantify the difference in physics captured by FFT coefficients using a K-S statistic test [33].

5.1 Learnability of the Analysis

Table 3. K-S Statistic Results: FC1 – FC5 : Failure Class I – V. W : Working Sensor

Failure	FC1	FC2	FC3	FC4	FC5	W
K-S Test	0.41	0.33	0.49	0.65	0.95	0.09

We performed a 2-sample **Kolmogorov–Smirnov (K-S)** statistic test [33] to validate our hypothesis that the A_{out} distributions from both working and failed sensors are different. We performed the tests at a significance level (α) = 0.05, that results in a K-S statistic threshold of 0.1 – a standard threshold for checking if sample differences in two distributions imply difference in population. Table 3 shows the K-S statistic (`kstest2` in Matlab [34]) computed for each class of failure relative to a working sensor in columns FC1 – FC5. We make the following observations :

Failed sensors have different frequency characteristics compared to a working sensor: Each failure class FC1 — FC5 has a K-S statistic value > 0.1 when compared to a working sensor (W), indicating that the distributions of faulty and working sensors differ.

The worse the failure, higher the K-S statistic: A sensor with oil condensation on the lens (C4) (a more pernicious fault leading to MO) resulted in a K-S statistic of 0.65 compared to 0.33 for a sensor with lens deformation (C2) (milder fault resulting in some blind spots). The K-S statistic captures the divergence between physics of failed and working sensors.

Working sensors have similar characteristics: We performed a sanity check between distributions of multiple working sensors and noticed the consistency among them in the frequency domain. Computing the K-S statistic resulted in a value below the 0.1 threshold implying similar physics as seen in Figure 8(f).

Summary: K-S test shows the possibility of information that can be extracted to generate a physics-based reliability feature vector for the PIR sensor. This also motivates using classification techniques to identify the information difference between different failure scenarios [9, 15].

5.2 Feature Selection and Importance

We use both the time domain and frequency domain features of the A_{out} signal to classify the sensor to either working or one of the faulty classes. As the number of features that can be derived is huge (e.g., in our case 305), we use the **Benjamini–Hochberg** (**B–H**) feature selection algorithm [8] to estimate and analyze *feature importance* in the A_{out} collected for working and each class of faulty sensors.

B-H technique is used offline with training data to decide which features are useful in predicting the label of the sensor (e.g., working vs. class X). The output of B-H feature selection process is a *feature importance score (F-score)* that indicates how useful or valuable each feature is in the decision making process. The higher the F-score, the more important the feature is to prediction.

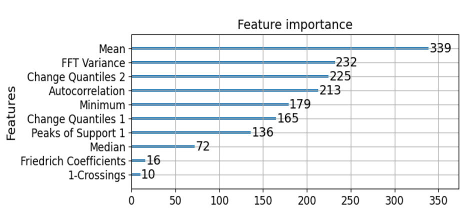
B-H feature selection process works by initially training an ensemble of decision trees on all features, derived from both time and frequency domain features as mentioned in Christ et al. [13] and implemented in the open source library `tsfresh` [11]. It then measures the prediction accuracy using every feature. This gives a high accuracy at the expense of overfitting. Thereafter, each feature vector is independently evaluated with respect to its significance for prediction using hypothesis testing, assigning it an F-score. It then iteratively prunes features having low F-scores, trains decision trees using these reduced features, and measures prediction accuracy. The process stops when a user-set threshold of accuracy is met or when all the combinations have been tested.

Applying the B-H process to our training data containing both faulty and working sensors, *we pruned 305 features for our entire dataset to obtain 10 features at a slightly better accuracy.* The larger set of features are derived from the list of features in Table 4. `tsfresh` attempts to identify the strongly and weakly-relevant features in the fault detection and diagnosis process. It uses an algorithm named `fresh` ("FeatuRe Extraction based on Scalable Hypothesis"), that has been shown to be an efficient, scalable feature extraction technique used in early stages of ML pipelines [13, 14]. `fresh` uses a 3-phase technique to select significant features while controlling the percentage of irrelevant features. First, `fresh` characterizes the time series using the larger set of features, some of which are indicated in Table 4. Note that the complete feature set is comprehensive and also includes composite features derived from the simple features. In the second step, each feature vector is individually and independently evaluated with respect to its significance using hypothesis testing. As a part of the hypothesis testing, the significance of each feature is quantified into a p-value, with smaller p-values indicating that the feature is not statistically significant. In the third and last step, the feature vector is evaluated using the B-H process to decide which features

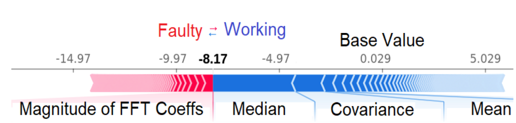
Table 4. Features Before Pruning, Calculated Using the `tsfresh` [11] Library

Feature	Description
abs_energy	Absolute energy i.e., sum of squares
agg_autocorrelation	autocorrelation of the time series
agg_linear_trend	linear least-squares regression aggregated over chunks versus the sequence from 0 up to (number of chunks - 1)
approximate_entropy	vectorized approximate entropy
c3	c3 statistics to measure non linearity
binned_entropy	calculates entropy over a set of equidistant bins
energy_ratio_by_chunks	the sum of squares of chunk expressed as a ratio with the sum of squares over the whole series
fft_aggregated	the spectral centroid (mean), variance, skew, and kurtosis of the absolute fourier transform spectrum
fft_coefficient	the fourier coefficients of the one-dimensional discrete fourier transform
fourier_entropy	the binned entropy of the power spectral density of the series
friedrich_coefficients	Coefficients of polynomial fitted to the Langevin model
linear_trend	a linear least-squares regression for the values of the series versus the sequence from 0 to length of the time series minus one
number_peaks	the number of peaks of at least support n
root_mean_square	the root mean square (rms) of the series
number_crossing_m	the number of crossings of series across m
sample_entropy	sample entropy of the series
variation_coefficient	the variation coefficient (standard error / mean, give relative value of variation around mean) of the series
ratio_beyond_r_sigma	ratio of values that are more than r standard deviations away from the mean
others	mean, variance, quantile, mean absolute change and so on

`tsfresh` uses hypothesis testing to prune and select most significant features.



(a)



(b)

Fig. 9. (a) Top 10 features selected using B-H procedure, (b) Summary of Shapley values showing FFT coefficients separate faulty and working classes.

to keep—by transforming the hypothesis testing scores into F-scores. This step ensures that incorrectly rejected features are regulated while conducting multiple hypothesis tests. Figure 9(a) plots the F-score for the selected 10 key features in both time and frequency domains. These include: (a) FFT Variance that indicates how spread the frequency distribution is, (b) Autocorrelation that generalizes periodicity or patterns in the time domain, and (c) Measures of central tendency such as mean, median and change quantiles that measure the time domain changes within a corridor. This process is performed offline and the significance of the pruned feature set is guaranteed by the B-H algorithm. Note that both Friedrich coefficients and Peaks of support capture short term dynamics of the signal. The mathematical definition of these features can be found in statistics textbooks and implementations in the `tsfresh` GitHub repository [11]. We can continue the

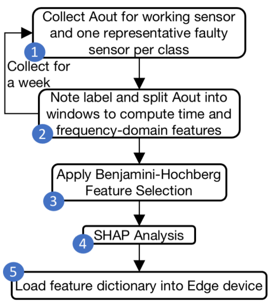


Fig. 10. Pre-deployment stage.

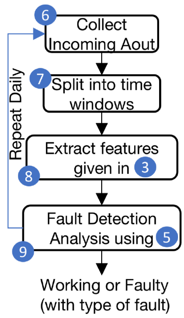


Fig. 11. Deployment stage.

Table 5. Details of Evaluation Setup Used in PIRMedic

Component	Specifications & Details
Data Collection & Analysis Platform	
Arduino Mega 2560 Rev3	ATmega2560 16 MHz with 256 kB Flash
Raspberry Pi v3.0 Model B	Quad Core 1.2GHz Broadcom BCM2837 64bit CPU, 1GB RAM
Connectivity & Power	
USB Serial Cable	Interface between Raspberry Pi and Arduino Microcontroller
Power Supply	5V for Arduino and 3.3V for Raspberry Pi
Sensor	
PIR Sensor	Murata PIR Evaluation Board IMX-060

feature pruning process by training decision trees on combinations of features from these 10 features and measuring prediction accuracy. We stopped the feature pruning with 10 features since we consistently obtained an accuracy of 97–99% in practice with this smaller set of features.

5.3 Monitoring Feature Performance Using Shapley Additive Explanations (SHAP)

Once the important features are identified using the B-H feature selection process, we use the technique of SHAP [30] values to *explain or interpret* the output of a prediction. SHAP uses game-theoretic techniques to explain how the output of a prediction changes by conditioning on every feature present in the reduced feature set given by B-H process.

Figure 9(b) shows an example of how SHAP values are analyzed for a particular prediction on a classification between faulty and working sensors. The features in red contribute to the sensor being predicted as faulty, whereas the features in blue contribute to the sensor being predicted as working. As seen, the competing features that contributed to this decision are the FFT coefficients and the statistical features *viz.*, mean, median, and covariance. In this prediction, the FFT coefficients push the prediction to that of a faulty sensor while the mean, median, and covariance push the prediction to a working sensor. The base value is the average SHAP value output by the model for the entire training dataset, and the value in bold is the SHAP value for this particular prediction. We leverage an open-source library implementation of the algorithm from Lundberg et al. [12, 30] that sorts the features by sum of SHAP value magnitudes over all the samples and conditionally rejects the feature contributing lowest to the prediction. Thus, we use the feature explanations from SHAP values to tune the model performance.

5.4 PIRMedic: Fault Detection and Analysis Algorithm

We now combine the insights gathered from physics and machine learning techniques to develop a unified algorithm, called PIRMedic, that is deployable at the edge, to isolate and diagnose faults. PIRMedic consists of two stages: (i) pre-deployment (training) stage, wherein the reference A_{out} values for a set of sensors (both working and faulty) are collected offline and analyzed to extract key features and (ii) deployment stage, wherein the features from a operational A_{out} signal is matched with the previously extracted features in order to detect and isolate faults. We next describe both the stages in detail.

5.4.1 Pre-Deployment Stage. We first collect A_{out} from a set of sensors, comprising both working and faulty ones such as the ones shown in Figure 4. The collected time series data are labeled to form a training dataset. We perform the B-H feature selection on this dataset to extract key features such as FFT coefficients of A_{out} for each class of sensor. This forms a feature dictionary containing a smaller number of features along with the failure as a label. We further verify the importance of these features using SHAP analysis and refine the features if necessary. We then build a classifier model to uniquely identify the sensor failure and load this model into the edge device connected to the PIR sensor.

We now describe the steps involved (Figure 10): 1 We deploy the working and faulty sensors for a short duration (say few days) in a real-world environment to collect realistic A_{out} signals for low, medium, and high occupancy. Note that this is a one-time activity performed for a specific PIR sensor type/manufacturer. We used 15 sensors comprising a mix of working and faulty sensors capturing failures in each class as our training set. 2 We note the label of a sensor and split the A_{out} values into equal-sized time slices (windows) to calculate different time and frequency-domain features. 3 We apply B-H feature selection process for each type of sensor to identify unique and key features. 4 We performed SHAP analysis to understand the performance of each feature toward classification and refine them accordingly. 5 We use the final set of features along with class labels to build a classifier model (we use random forest as our classifier model). This model is then loaded onto the edge devices for fault detection and analysis. We note that while 2 and 3 are computationally expensive, it is acceptable as it is performed offline only once in the pre-deployment.

5.4.2 Deployment Stage. This stage consists of the following steps (Figure 11): 6 First, the operational A_{out} output signals for the deployed sensors are collected. 7 We split the A_{out} into equal-sized windows using same window size as in the pre-deployment. 8 We extract the necessary features from the A_{out} signal. 9 We use a classifier to (a) isolate faulty from working sensors and (b) identify the class of failure in the faulty sensor if applicable. The fault detection (steps 6 – 9) depends on the application requirements, i.e., every hour, day, or week.

6 Implementation

Our solution is designed using **commodity off the shelf** (**COTS**) components. The open-source implementation[2] includes:

(a) **Edge Hardware** consisting of – (i) a PIR sensor [41], (ii) a bare-metal platform *viz.,* Arduino Mega **Microcontroller Unit** (**MCU**) [48] for sensor data capture, and (iii) a Linux-based platform *viz.,* Raspberry Pi [45] for executing the fault detection and diagnosis algorithm, PIRMedic. The specifications of hardware used can be seen in Table 5.

The Arduino MCU polls the sensor hardware signals (i.e., C_{out} and A_{out}) periodically over a GPIO interface. We poll the PIR sensors at a frequency ≥ 20 Hz due to the *Nyquist criterion* as the human motion information in the PIR sensor is between $0 - 10$ Hz. Note that using a bare-metal platform allows us to capture A_{out} samples at a constant rate. The Raspberry Pi platform performs the fault detection and analysis for the data captured at the Arduino MCU. The Raspberry Pi and Arduino are connected via a Micro-USB cable and the data captured at the Arduino MCU is serially transferred to the Raspberry Pi for analysis, as shown in Figure 12.

(b) **Edge Implementation** We used standard python libraries such as `tsfresh` [11, 14] for computing features such as FFT and other statistical features as a part pf feature extraction, `numpy` for data preprocessing and `xgboost` [12] for implementing the machine learning classifiers. The end-to-end workflow of our solution is illustrated in Figure 13.

6.1 Parameters of PIRMedic

(i) **Size of Window**: The collected A_{out} waveform is split into equal-size sample windows over which features are calculated. We observed that using a window too small (e.g., 128) does not allow

[2] Available at: https://github.com/synercys/PIRMedic

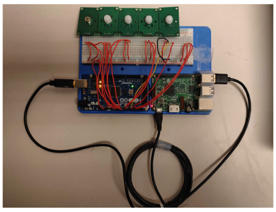


Fig. 12. Evaluation platform showing PIR sensor connected to Raspberry Pi and Arduino.

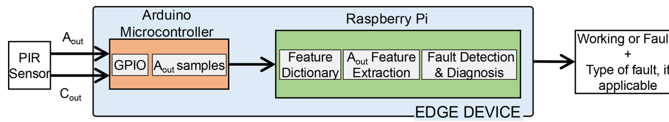


Fig. 13. PIRMedic workflow. The PIR sensor is connected to an Arduino microcontroller via a GPIO interface. An ADC onboard the microcontroller converts A_{out} and C_{out} values into discrete samples, that are sent to a Raspberry Pi for fault detection and diagnosis.

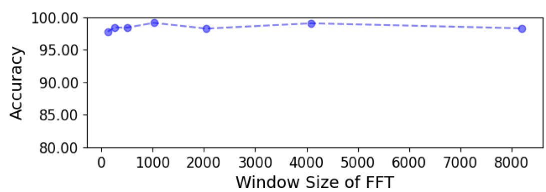


Fig. 14. Model accuracy for different window sizes. We choose 1024 as the default window size.

us to capture sufficient features for a failure type and could lead to a large number of false alarms. Also, too large of a window size (e.g., 8192) leads to an overlap in the features of multiple failure classes leading to a loss of accuracy. The variation of model accuracy as a function of window size is shown in Figure 14. We chose a window size of 1024 samples as it gave us a good tradeoff between time to capture a window (approx. 50 seconds) and accuracy (> 98%).

(ii) **Benjamini-Hochberg (B-H) Feature Selection** While there are more than 300 features that can be derived for time series analysis of A_{out}, B-H feature selection process (derived from parameterized hypothesis testing) [8, 13] prunes this to a lower number of features that can sufficiently capture the physics of the system. We observed that using more than 10 features did not contribute to a significant increase in accuracy (beyond 98%) and can, in practice, lead to lower performance due to overfitting as described in Section 5.2.

(iii) **Classifier Model**: We used a Random Forest classifier [9, 42] owing to the capability to classify data based on "entropy" or the information gain. Random Forest uses an ensemble of trees and has been shown to give high predictive accuracy and control over-fitting in practice, both of which are issues with decision trees. Additionally, Random Forests are amenable to interpretation by techniques such as Shapley TreeExplainer [29] that explains the features influencing prediction. *Finally, we note that A_{out} is compatible with any classification algorithm.*

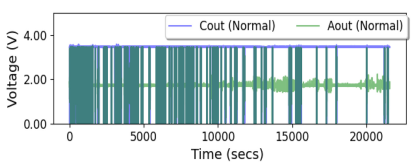
7 Evaluation

The evaluation is intended to answer the following questions—(a) can we detect failures i.e., separate faulty PIR sensors from working ones? and (b) what can we learn about the failure? In other words, can we perform diagnosis that can aid in replacing or repairing the sensor? We answer these questions in practical deployments, each consisting of the following stages:

(a) **Stage I: Failure Detection** The goal of this stage is to isolate defective, faulty sensors from functional, working sensors. Note that we do not diagnose the failure in this stage i.e., the sensors deemed faulty could be so due to multiple, as yet unknown reasons.

(b) **Stage II: Failure Diagnosis** In this stage, we diagnose the failure by mapping it to the taxonomy defined in Section 3.3. Using the taxonomy as a reference, we seek to conclude whether it is the lens, pyroelectric or electronics subsystem that contains the failure.

Fig. 15. Deployment scenarios used for evaluating PIRMedic in PIR sensor failure detection and diagnosis *viz.*, elevator, lobby of a building and Starbucks coffee shop.



(a)

Fault Present	WIN = 2432	
	MO	FD
Lens Dislodged (A)	343	50
Lens Obstruction with Black Tape (B)	529	0
Pyroelectric Element Heat Damage (D)	247	55
Optical Filter Covered (E)	526	0
Total Sensors Deployed (5)		

A – Lens Dislodged
B – Lens Obstruction with Black Tape
C – Working Sensor
D – Pyroelectric Element Heat Damage
E – Optical Filter Covered

True \ Predicted	A	B	C	D	E
A	1	0	0	0	0.003
B	0	1	0	0	0
C	0.003	0	0.99	0.003	0
D	0	0	0	1	0
E	0	0	0	0	1

(b)

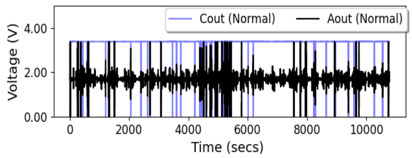
Fig. 16. Elevator deployment (a) Occupancy: The graph is a 6 hour deployment during business hours, (b) Deployment statistics and confusion matrix of the classification model. Column titles: **WIN**→Total Number of Windows, **MO**→Misses Obstacle, **FD**→ False Detection.

(c) **Stage III: Fine-Grained Fault Analysis** In this stage, we learn details about the failure as precisely as possible. For example, if we attribute a fault to be Class III failure, we seek to narrow it down to see if it is due to deposition of dust or paper on the lens.

7.1 Real-World Deployments

We performed the following *deployments across multiple scenarios and geographic locations, viz.,* (a) in the elevator of a building (Section 7.1.1), (b) in the lobby of a university building (Section 7.1.2), (c) in a Starbucks coffee shop (Section 7.1.3), and (d) in a cafeteria (Section 7.1.4) (India). The different deployment scenarios in (a) –(c) are shown in Figure 15, and aim to capture diverse environmental conditions and challenges (heat, dust, humidity etc.). We used 15 sensors in total across all our deployments comprising of five working sensors and two faulty sensors per class. A combination of these was used in each of our deployments. In each of these deployments, we followed the pre-deployment process, as mentioned in Section 5.4.1, where different types of faulty sensors were deployed for a week to collect its A_{out} and C_{out}. This pre-deployment is a one-time process to calculate different time and frequency domain characteristics capturing the physics of A_{out}. A subset of these sensors used in the pre-deployment was randomly picked in each of the deployments. These sensors were deployed (Figure 4) over a duration of 3 months and collected data both during times of low occupancy (weekends, late nights) and high occupancy (weekdays, working hours). Note that no auxiliary private data such as audio or video were captured during these deployments.

7.1.1 Deployment in a University Elevator (US). We deployed working and faulty sensors along the inside wall of an elevator in our university research building (left Figure 15). The data collection captured scenarios of both high and low occupancy during and outside of business hours. In addition to faulty sensors, we placed a normal, working sensor to keep track of the actual motion in the elevator. Figure 16(a) plots C_{out} and A_{out} values captured by the working sensor showing the



(a)

Fault Present	WIN = 935	
	MO	FD
Lens Dislodged (A)	608	15
Lens Deformed (B)	13	2
Pyroelectric Element Oil Damage (C)	20	15
Total Sensors Deployed (4)		

A – Lens Dislodged
B – Lens Deformed
C – Pyroelectric Element Oil Damage
D – Working Sensor

True \ Predicted	A	B	C	D
A	0.91	0.001	0.083	0.005
B	0.001	0.97	0.003	0.025
C	0.08	0.012	0.85	0.06
D	0.004	0.05	0.04	0.9

(b)

Fig. 17. Lobby deployment (a) Occupancy graph during late evening from 6.45 pm to 9.45 pm at the lobby of our university building, (b) Deployment statistics and confusion matrix of the classification model. Column titles: **WIN**→Total Number of Windows, **MO**→Misses Obstacle, **FD**→ False Detection.

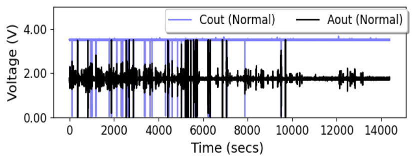
actual traffic in the elevator on one of the days.Vertical stripes on the graph showing oscillations of C_{out} (in blue) are points of occupancy i.e., denoting a person's entry/exit at the elevator. Likewise, regions of graph with no occupancy is represented by a flat line, hovering around 3.7 V.

For faulty sensors, we measure the number of time windows where an obstacle was captured or missed. The performance of faulty sensors relative to a working sensor is tabulated in Figure 16(a). When the lens cap is dislodged, it misses obstacles in 343 time windows whereas the (thermal) breakage of the pyroelectric element results in 247 misses. Overall, we observe that MO are more common than false alarms. This ties into common observations in conference rooms operated with PIR sensor-controlled lighting, where lights turn off even with occupants present.

As described in Section 5.4, PIRMedic collects the A_{out} signals and uses a machine learning model on their extracted features detect faulty sensors. Using a random forest classifier, we obtain the confusion matrix (Figure 16(b)), which shows that predicted label matches the true label for each failure. We note that – (a) the working sensor has distinct characteristics compared to the faulty sensors and is isolated correctly, (b) faulty sensors with different failures in lens and pyroelectric subsystem are identified to the failure class correctly as observed by diagonal of the confusion matrix. This implies that the features of A_{out}, derived by PIRMedic, are able to differentiate between lens being dislodged, lens being covered with tape, oil condensation on the optical filter, and heat damage on the pyroelectric element.

7.1.2 Deployment in a University Lobby (US). We deployed our sensors in the first floor lobby space of our university building (center Figure 15). The lobby deployment is noisier with fewer constraints on direction of motion, compared to the confined space of an elevator. As in the elevator deployment, we deployed a combination of working and faulty sensors shown in the table in Figure 17(b).

Figure 17(a) shows a 3 hour slice of the deployment. The vertical stripes on C_{out} correspond to traffic entering/exiting the lobby in the field of view. The performance of the sensors deployed, in terms of FD and MO, is shown in the table in Figure 17(b). The sensors with lens dislodged missed a large amount of obstacles. This is due to the lack of focus of thermal radiation on the pyroelectric element resulting in dispersion of radiation due to the large lobby area. As a result of this failure, only obstacles that line up in the reduced field of view are captured. This issue is not present in an elevator due to its confined space. Other faults such as lens deformation and damage due to oil condensation experienced lesser number of misclassifications. Upon extracting the features of the A_{out} signals, and classifying it using the procedure in Section 5.4, we observe the confusion matrix as shown in Figure 17(b). The high fraction along the diagonal shows that a majority of sensor failures are classified and identified correctly, as the features are able to bring out differences in the physics of failures. Note that we observe some misclassifications of sensor faults (e.g., for C). We argue that this is due to the deployment area of a lobby being open and the limited coverage area of the sensor, both of which can lead to imperfect capture at the periphery.



(a)

Fault Present	WIN = 226	
	MO	FD
Lens Dislodged	27	0
Lens Deformed	35	0
Lens Covered with Dust	26	0
Pyroelectric Element Oil Damage	8	0
Sensor Board Electronic Fault	226	0
Total Sensors Deployed (6)		

A – Lens Dislodged
B – Lens Deformed
C – Dust on Lens
D –Pyroelectric Element Oil Damage
E – Working Sensor

True \ Predicted	A	B	C	D	E
A	1	0	0	0	0.003
B	0	0.99	0	0.007	0.003
C	0	0.025	0.95	0.02	0.007
D	0	0.011	0	0.98	0.007
E	0	0	0	0.007	0.99

(b)

Fig. 18. **Starbucks deployment occupancy**: Evening deployment from 4.45 pm to 8.45 pm. The second half has much lesser little foot traffic. (b) Deployment Statistics and confusion matrix of the classification model. Column titles: **WIN**→Total Number of Windows, **MO**→Misses Obstacle, **FD**→ False Detection.

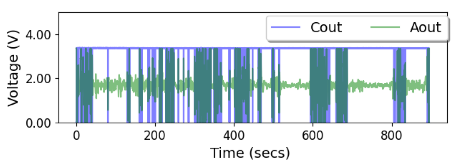


Fig. 19. **Cafeteria deployment occupancy** during lunchtime. A 15 min. slice of traffic at the checking counter.

Training (MB)	**32**	**64**	**96**	**112**	**128**
Accuracy (%)	92.4	95.8	98.6	99.1	99.4

Fig. 20. **Failure detection accuracy** as a function of the size of training dataset

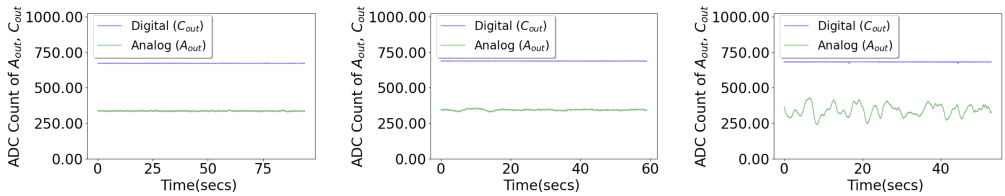
Consequently, for traffic at the edges of field of view, this reduced thermal radiations could lead to some misclassifications.

7.1.3 Deployment in a Starbucks (US). We deployed PIR sensors near the ordering queue of the coffee shop during business hours to capture the typical foot traffic (right Figure 15). Our deployment captured linear, regularized bidirectional movement (entry followed by exit of the region of interest) (Figure 18(a)). Our deployment lasted for a duration of 4 hours on a busy Friday evening. We observed that out of 226 time windows (each containing 1024 samples), there were MO across: (i) 27 windows in class I, (ii) 35 windows in class II, (iii) 26 windows in class III, and no windows in class IV. The MO and FD (none in this case) are summarized in Figure 18(b). Using PIRMedic, we observe high fractions along the diagonal of the confusion matrix; hence the predicted label of sensor failure matches the true label.

Furthermore, despite the oil impurity not being enough cause it to alter performance, we still classify it is a separate class from the physics point of view due to its different behavior of A_{out} from that of a working sensor. Thus, in line with the previous deployments, we can use the features of A_{out} to form a signature of the physics.

7.1.4 Deployment in a Cafeteria. To evaluate the analysis of A_{out} during quiet times, we deployed a mix of three types of sensors with lens failures—lens fallen off, lens covered, and some working sensors during the lunch hour at a cafeteria in India. The traffic during lunchtime in the cafeteria is as shown in Figure 19. As these were lens-related failures, we looked at the time-domain characterization of A_{out} during quiet times to understand the status of lens. The time domain representation of A_{out}, during a period of no occupancy, is as shown in Figure 21. The y-axis is plotted in terms of the 10-bit ADC representations of voltage to highlight the differences. For a 10-bit ADC, 1023 units (i.e., $2^{10} - 1$) correspond to a voltage of 5 V. Therefore, a voltage of 1.8 V corresponds to 340 units.

We note visually that the noise floor at output of the lens subsystem is higher with the lens dislodged (Figure 21(c)) as compared to both a normal sensor (Figure 21(b)) and one with lens covered (Figure 21(a)). Specifically, we observed a variance of 10 $Volt^2$ when lens was covered, 250 $Volt^2$



(a) Lens Covered Completely (b) Lens in Place (Working) (c) Lens Dislodged

Fig. 21. The variance in A_{out} during known quiet times contains hints about lens failures in the PIR sensor.

when lens was in place and working, and 1500 $Volt^2$ when lens had fallen of completely. Consequently, a low threshold (T_L) of 100 $Volt^2$ and a high threshold (T_H) of 1000 $Volt^2$ was sufficient to separate between the three types of sensors. Note that the variance had to be computed during a quiet period of no traffic in this case.

We now discuss the different stages of PIRMedic, as enumerated in the beginning of this section.

7.2 Stage I – Failure Detection

	Precision	Recall	F1-score	Support
Faulty	0.99	1.00	0.99	1958
Working	1.00	0.98	0.99	1450
Accuracy			0.99	3408
Macro Avg	0.99	0.99	0.99	3408
Weighted Avg	0.99	0.99	0.99	3408

Classifier Parameters	Remarks
Type	Random Forest
Classification Criterion	Entropy
Estimators	10

Fig. 22. Fault detection showing that faulty sensors isolated from working sensors by extracting FFT of A_{out} and using a Random Forest Classifier.

We aggregated all the sensors (both working and faulty) to *separate faulty sensors (without specifying which class) from working sensors*. We collected FFT coefficients of A_{out} on a pre-deployment lasting 1 week and performed daily collection of A_{out} signals from all the deployed sensors. We used a Random Forest classifier with specifications shown in Figure 22. Once predicted, we manually verified the true status of the sensor and observed an accuracy of over 99% in predicting if the sensor is working or faulty. Thus, FFT-based features of A_{out} were able to capture the physics, and hence separate faulty sensor physics from working sensor physics. We also varied the *size of training dataset* (128 MB corresponds to 1 week pre-deployment) to show the robustness of failure detection in Figure 20.

7.3 Stage II – Failure Diagnosis

Diagnosing the *cause of failures (i.e., class)* requires further investigation of A_{out} from faulty sensors. We combine our understanding of the PIR sensor i.e., physics from Section 3.1, failure taxonomy from Section 3.3 and the frequency analysis of failures from Section 4.5 and feature analysis from Sections 5.2–5.3. Figure 23(a) plots the CDF of FFT coefficients obtained for working sensors and various faulty sensors. The figure shows that although the distribution of working sensors (bottom-most curve) is distinct from failed sensors, different classes of failures have distributions that are not easily separable. Thus, relying on merely the FFT is insufficient. Consequently, we look at an additional 10 features as described in Sections 5.2–5.3 and shown in Figure 9(a).

We validated this across the multiple deployments in the wild — Elevator (Section 7.1.1), Lobby (Section 7.1.2), Starbucks (Section 7.1.3), and Cafeteria (Section 7.1.4). Our real-world measurements demonstrate that features of A_{out} extracted during training and B-H feature selection is robust to variations in shapes and sizes of obstacles.

7.4 Stage III – Fine-Grained Failure Analysis

In fine-grained fault analysis, we point to the *precise reason* for the fault where applicable. For example, in Class III faults it is useful to distinguish between different foreign substances contaminating the lens that can cause poor performance.

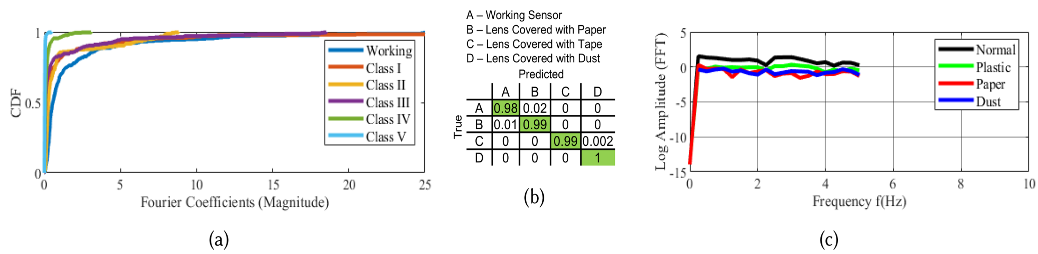


Fig. 23. (a) CDF of FFT coefficients showing different distributions depending on the class of faults, (b) Fine-grained fault analysis, (c) $A_{\mathbf{out}}$ waveforms showing different behavior for paper, plastic, and dust obstructions.

To investigate this, we selected the set of faulty sensors identified as Class III faults in Stage II: some covered with paper, some with plastic tape and some with dust and observe the A_{out} output closely. Although all the failure scenarios result in the obstacle being missed, frequencies present in A_{out} are different in each case, as shown in Figure 23(c), due to the distinct thermal absorption characteristics of the materials. Hence, we observe that the A_{out} features can separate Class III failures into different fine-grained sub-classes as shown in Figure 23(b).

It is to be noted that fine-grained fault analysis does not work in all cases. For example, differentiating between a lens puncture (a hole in the lens) and a lens deformation (defect in lens curvature), both examples of Class II failures, is complex. This is because both lens puncture and lens deformations lead to blind spots in the frequency spectrum. Thus, we need to use knowledge of the subsystem physics before applying the fine-grained analysis.

Summary: (i) The FFT representation of A_{out} is used to perform binary classification between working and faulty sensors, (ii) Considering multiple direct and derived time and frequency domain signal characteristics of A_{out} can diagnose the cause of failure in the sensor and (iii) Fine-grained fault analysis is possible in select failure cases (e.g., Class III) by reanalyzing the A_{out} waveforms.

8 Discussion

Practicality: We note A_{out} is an intrinsic signal that is present in all PIR sensors, since it is part of the sensing physics in a PIR sensor. However, as most applications require only binary occupancy data (i.e., occupied or not occupied), some commercial-grade PIR sensors typically do not provide easy access to the A_{out} signal. While PIRMedic focuses on low-cost, commodity PIR sensors costing less than \$10-15, its physics is identical to that seen in commercial-grade PIR sensors. In fact, we obtained one such commercial PIR sensor from our building ceiling and analyzed the A_{out} waveform by connecting it to an oscilloscope. In our preliminary observations, we saw that the shape of A_{out} is similar to that in the low-cost sensors used in this article. Additionally, there exist some commercial sensors (e.g., [39, 40]) that expose A_{out}. In addition, existing sensors can be modified to expose A_{out}— most do as a test point on the PCB. This modification would not incur any additional hardware costs, with the exception of a new pin connector. We believe that our work can provide further incentive to other manufacturers. Lastly, our conference paper in ACM BuildSys [23] was added to the product webpage by Murata electronics [35] and we hope other manufacturers use this to expose A_{out} in their sensor designs.

Latency: PIRMedic can precisely determine if a sensor is working or faulty in single time window, which in our case comprised of 1024 samples. Since we sample at 20 Hz, the end-to-end process of data acquisition, processing, and inference can be performed in a little over one minute.

Consistency of behavior: We observed that the inherent characteristics of the A_{out} signal, both in time and frequency domains, remain consistent across different manufacturers. This is expected since all PIR sensors use a pyroelectric element and the physics of operation is deterministic.

Scale of Deployment: Our deployment comprised of 15 sensors across various building occupancy scenarios, both in India and US, where we used a random combination of sensors in each of these deployment scenarios. While we acknowledge that our deployment scale is small-scale from the perspective of a city or a municipality spanning multiple buildings, the techniques described in this article can be applied to a larger scale deployment comprising more sensors. To scale to larger deployments, one needs to—(a) have edge platforms (Raspberry Pi and Arduino) located in close proximity to each of the sensor deployments (e.g., in the aisle of a building for a series of rooms), (b) run wiring from each of those sensors to the edge platforms (e.g., one can leverage points where PIR sensors are connected to lighting and HVAC systems) and (c) perform computation on a server on the premises of the building, such as the BMS, to ease the computational load on the edge platforms.

Differentiation of Faults: The goal of PIRMedic is to *associate each faulty sensor to a failure in the corresponding sensor subsystem* shown in Table 1. The classification of faults in this manner was a deliberate choice – based on the physics of the PIR sensor, as we wanted our solution to be specific to sensors that are ubiquitous in modern buildings. Additionally, while a PIR sensor can have multiple faults present, we intentionally study faults in isolation to separate their physics and understand the impact of each of these failures on individual subsystems.

Degradation and Aging: PIR sensors are susceptible to environmental elements over time, leading to degradation and aging. While we previously examined one form of degradation in Section 4.2, where dust accumulation on the PIR sensor was studied, we plan to investigate additional forms of aging and degradation in the future to analyze their impact on sensor physics and the fault diagnosis process. The primary goal of this work was to understand failures and their effects on sensing physics; therefore, degradation and aging were excluded from the current analysis. However, as deployments become more long-term, these factors may play a crucial role in the analysis.

Design: We envision PIRMedic to be connected to a BMS and provide real-time fault detection. We envision developing APIs for application developers to build higher-level diagnostic tools on top of PIRMedic.

Future work and research challenges: Gupta et al. [17] in the ACM LCTES'19 keynote mention the importance of fusing and navigating multiple data streams and contextualizing data streams in a CPS such as a BMS. They emphasize the importance of capturing taxonomy of entities (e.g., data points, location, and equipment) and their relationships for developing building applications. Additionally, schema systems such as BRICK [7], Scrabble [25], and Plaster [26] are designed as solutions developed to contextualize data streams using meta-data. One could envision using PIRMedic to enable fault diagnosis and detection in conjunction with these building-based schema systems thereby making reliability a key facet of such programming platforms. Almanee et al. [3] develop a middleware, SemIoTic, that provides a semantic view of an IoT space to both the inhabitants and application developers. SemIoTic translates user-defined requests into underlying actions and can also be enhanced with the failure information from PIRMedic. Koh et al. [27] demonstrate algorithms to standardize building metadata which can become a key enabler in deploying smart building applications over heterogeneous buildings. Their key contribution is a graphical user interface with a well-defined workflow built upon a common programming interface, called Plaster. We envision that PIRMedic can enable the integration of fault detection and diagnosis into building metadata-based schema systems. This will also improve the robustness of various downstream

ML pipelines comprising the BMS that rely on the correctness and validity of data from building sensors such as the PIR sensor.

9 Conclusion

The approach of using physics in conjunction with machine learning techniques can pave way for a dependable and reliable IoT. The analysis of intrinsic signals allows us to better understand the physics of a sensor that can aid in fault detection, diagnosis, and maintenance of sensors. We show that in the case of PIR sensors, the appropriate use of intermediate analog signals offers potential to perform in-situ fault diagnosis thereby reducing costs and improving reliability. Due to lack of standard designs of PIR sensors, we propose that sensor vendors include the A_{out} signal in all PIR sensor designs to aid in the reliability of PIR sensors.

Acknowledgments

We would like to thank Denizhan Kara (UIUC) and Kyo Kim (UIUC) for their insights on feature analysis.

References

[1] Anshul Agarwal. 2021. A novel approach to save energy by detecting faulty HVACs. *Journal of The Institution of Engineers (India): Series B* 103 (2021), 305 – 311. Retrieved from https://api.semanticscholar.org/CorpusID:238649798

[2] Anshul Ajay Agarwal and Krithi Ramamritham. 2022. A novel approach for deploying minimum sensors in smart buildings. *ACM Transactions on Cyber-Physical Systems* 6 (2022), 2:1–2:29. Retrieved from https://api.semanticscholar.org/CorpusID:244564069

[3] Sumaya Almanee, Georgios Bouloukakis, Daokun Jiang, Sameera Ghayyur, Dhrubajyoti Ghosh, Peeyush Gupta, Yiming Lin, Sharad Mehrotra, Primal Pappachan, Eun-Jeong Shin, et al. 2019. Demo abstract: SemIoTic: Bridging the semantic gap in IoT spaces. In *BuildSys 2019-6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation.*

[4] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge Luis Reyes-Ortiz. 2013. A public domain dataset for human activity recognition using smartphones.. In *Proceedings of the International European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. Vol. 3. 3.

[5] Application Note of Sensor & Lens. Application Note Murata PIR Sensor. Retrieved from https://www.murata.com/~/media/webrenewal/products/sensor/infrared/applinote_pir.ashx?la=ja-jp. (n.d.).

[6] H. Ayadi, A. Zouinkhi, B. Boussaid, and M. N. Abdelkrim. 2015. A machine learning methods: Outlier detection in WSN. In *Proceedings of the 2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA'15).* 722–727. DOI: https://doi.org/10.1109/STA.2015.7505190

[7] Bharathan Balaji, Arka Bhattacharya, Gabriel Fierro, Jingkun Gao, Joshua Gluck, Dezhi Hong, Aslak Johansen, Jason Koh, Joern Ploennigs, Yuvraj Agarwal, et al. 2016. Brick: Towards a unified metadata schema for buildings. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments.* 41–50.

[8] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)* 57, 1 (1995), 289–300.

[9] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[10] Tusher Chakraborty, Akshay Uttama Nambi, Ranveer Chandra, Rahul Sharma, Manohar Swaminathan, Zerina Kapetanovic, and Jonathan Appavoo. 2018. Fall-Curve: A novel primitive for IoT fault detection and isolation. In *Proceedings of the 16th ACM SenSys.* 95–107. DOI: https://doi.org/10.1145/3274783.3274853

[11] Maximilian Christ. 2018. tsfresh. Retrieved from https://github.com/blue-yonder/tsfresh. (2018).

[12] Maximilian Christ. 2020. SHAP XGBoost Implementation. Retrieved from https://github.com/slundberg/shap. (2020).

[13] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. 2018. Time series feature extraction on basis of scalable hypothesis tests (tsfresh–A Python package). *Neurocomputing* 307 (2018), 72–77.

[14] Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. 2016. Distributed and parallel time series feature extraction for industrial big data applications. arXiv:1610.07717. Retrieved from https://arxiv.org/abs/1610.07717

[15] Padraig Cunningham and Sarah Jane Delany. 2020. k-Nearest neighbour classifiers–. arXiv:2004.04523. Retrieved from https://arxiv.org/abs/2004.04523

[16] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel. 2016. Blind mobile sensor calibration using an informed nonnegative matrix factorization with a relaxed rendezvous model. In *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'16).* 2941–2945. DOI: https://doi.org/10.1109/ICASSP.2016.7472216

[17] Rajesh K. Gupta, Jason Koh, and Dezhi Hong. 2019. New models and methods for programming cyber-physical systems (keynote). In *Proceedings of the 20th ACM SIGPLAN/SIGBED International Conference on Languages, Compilers, and Tools for Embedded Systems*. Retrieved from https://api.semanticscholar.org/CorpusID:184483241

[18] Tanmaey Gupta, Shubhankar Handa, and Akshay Nambi. 2023. Verified telemetry: A general, easy to use, scalable and robust fault detection SDK for IoT sensors. In *Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation (IoTDI'23)*. ACM, New York, NY, USA, 381–395. DOI: https://doi.org/10.1145/3576842.3582386

[19] Steve Hanly. Why the Power Spectral Density (PSD) Is the Gold Standard of Vibration Analysis. (n.d.). Retrieved from https://blog.endaq.com/why-the-power-spectral-density-psd-is-the-gold-standard-of-vibration-analysis#Benefits

[20] Hubbell. 2020. Ceiling Hard Wired Occupancy Sensor, 2,000 sq ft Passive Infrared, Ultrasonic, Office White. Retrieved from https://www.hubbell.com/hubbell/en/Products/Electrical-Electronic/Wiring-Devices/Lighting-Controls/CeilingWall-Sensors/ATD2000CRP/p/1545054?location=&brandCode=WDK. (2020).

[21] Hubbell. 2020. Wall Hard Wired Occupancy Sensor, 1,600 sq ft Passive Infrared, Ultrasonic, Office White. Retrieved from https://www.hubbell.com/wiringdevice-kellems/en/Products/Electrical-Electronic/Wiring-Devices/Lighting-Controls/CeilingWall-Sensors/ATD1600WRP/p/1545050. (2020).

[22] Brent Huchuk, Farid Bahiraei, and Saptak Dutta. 2021. Data-driven identification of occupant-thermostat interactions in small commercial buildings. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 341–344.

[23] Ashish Kashinath, Sibin Mohan, Akshay Nambi, and Sumukh Marathe. 2021. PIRMedic: Physics-driven fault diagnosis for PIR sensors. In *Proceedings of the 8th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation (BuildSys'21)*. ACM, New York, NY, USA, 71–80. DOI: https://doi.org/10.1145/3486611.3486658

[24] Nodira Khoussainova, Magdalena Balazinska, and Dan Suciu. 2006. Towards correcting input data errors probabilistically using integrity constraints. In *Proceedings of the 5th ACM International Workshop on Data Engineering for Wireless and Mobile Access (MobiDE'06)*. ACM, New York, NY, USA, 43–50. DOI: https://doi.org/10.1145/1140104.1140114

[25] Jason Koh, Bharathan Balaji, Dhiman Sengupta, Julian McAuley, Rajesh Gupta, and Yuvraj Agarwal. 2018. Scrabble: Transferrable semi-automated semantic metadata normalization using intermediate representation. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys'18)*. ACM, New York, NY, USA, 11–20. DOI: https://doi.org/10.1145/3276774.3276795

[26] Jason Koh, Dezhi Hong, Rajesh Gupta, Kamin Whitehouse, Hongning Wang, and Yuvraj Agarwal. 2018. Plaster: An integration, benchmark, and development framework for metadata normalization methods. In *Proceedings of the 5th Conference on Systems for Built Environments (BuildSys'18)*. ACM, New York, NY, USA, 1–10. DOI: https://doi.org/10.1145/3276774.3276794

[27] Jason Koh, Kuo Liang, Yiming Yang, Dezhi Hong, Yuvraj Agarwal, and Rajesh Gupta. 2019. Interactive building metadata normalization. In *Proceedings of the 6th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 340–341.

[28] Junxiu Liu, Yongchuang Huang, Yuling Luo, Jim Harkin, and Liam McDaid. 2019. Bio-inspired fault detection circuits based on synapse and spiking neuron models. *Neurocomputing* 331 (2019), 473–482. DOI: https://doi.org/10.1016/j.neucom.2018.11.078

[29] Scott M. Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M. Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. 2020. From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence* 2, 1 (2020), 2522–5839.

[30] Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30, Curran Associates, Inc., 4765–4774. Retrieved from https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

[31] B. Maag, Z. Zhou, and L. Thiele. 2018. A survey on sensor calibration in air pollution monitoring deployments. *IEEE Internet of Things Journal* 5, 6 (2018), 4857–4870. DOI: https://doi.org/10.1109/JIOT.2018.2853660

[32] Sumukh Marathe, Akshay Nambi, Manohar Swaminathan, and Ronak Sutaria. 2021. CurrentSense: A novel approach for fault and drift detection in environmental IoT sensors. In *Proceedings of the International Conference on Internet-of-Things Design and Implementation*. Retrieved from https://www.microsoft.com/en-us/research/publication/currentsense-a-novel-approach-for-fault-and-drift-detection-in-environmental-iot-sensors-2/

[33] Frank J. Massey Jr. 1951. The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association* 46, 253 (1951), 68–78.

[34] MATLAB Documentation Mathworks. 2015. kstest2: Two Sample Kolmogorov-Smirnov Test. Retrieved from https://www.mathworks.com/help/stats/kstest2.html. (2015).

[35] muRata. 2017. *muRata IMX-070 Pyroelectric Infrared Sensor Evaluation Board*. Retrieved December 23, 2024 from https://device.report/manual/9379538

[36] J. Murphree. 2016. Machine learning anomaly detection in large systems. In *Proceedings of the 2016 IEEE Autotestcon*. 1–9. DOI: https://doi.org/10.1109/AUTEST.2016.7589589

[37] Sujay Narayana, R. Venkatesha Prasad, Vijay S. Rao, T. V. Prabhakar, Sripad S. Kowshik, and Madhuri Sheethala Iyer. 2015. PIR sensors: Characterization and novel localization technique. In *Proceedings of the 14th ACM IPSN*. 142–153. DOI: https://doi.org/10.1145/2737095.2742561

[38] Online Forum on Fresnel Lens. PIR Fresnel Lens, Infrared Fresnel Lens. Retrieved from https://3dlens.com/pir-fresnel-lens.php. (n.d.).

[39] Panasonic. 2021. AMN22111 Industrial Devices. Retrieved from https://bit.ly/3Al6lvU. (2021).

[40] Panasonic. 2021. AMN23111 Industrial Devices. Retrieved from https://bit.ly/2VSawjM. (2021).

[41] PIR Evaluation Board Quick Start Guide. PIR Sensor Evaluation Board (IMX-070, IMX-060), Murata. Retrieved from https://bit.ly/2Y8Dj51. (n.d.).

[42] Robi Polikar. 2012. Ensemble learning. In *Ensemble Machine Learning*. Springer, 1–34.

[43] Alexander Power and Gerald Kotonya. 2019. Providing fault tolerance via complex event processing and machine learning for IoT systems. In *Proceedings of the 9th International Conference on the Internet of Things (IoT'19)*. ACM, New York, NY, USA, Article 1, 7 pages. DOI: https://doi.org/10.1145/3365871.3365872

[44] Haroon Rashid, Pandarasamy Arjunan, Pushpendra Singh, and Amarjeet Singh. 2016. Collect, compare, and score: A generic data-driven anomaly detection method for buildings. In *Proceedings of the 7th International Conference on Future Energy Systems Poster Sessions*. 1–2.

[45] Raspberry Pi Store Page 2021. Raspberry Pi 3 Model B. Retrieved from https://www.raspberrypi.org/products/raspberry-pi-3-model-b/. (Jan. 2021).

[46] Olga Saukh, David Hasenfratz, and Lothar Thiele. 2015. Reducing multi-hop calibration errors in large-scale mobile sensor networks. In *Proceedings of the 14th ACM IPSN*. 274–285. DOI: https://doi.org/10.1145/2737095.2737113

[47] Abhishek B. Sharma, Leana Golubchik, and Ramesh Govindan. 2010. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks* 6, 3, Article 23 (June 2010), 39 pages. DOI: https://doi.org/10.1145/1754414.1754419

[48] Arduino Store. 2021. Arduino Mega 2560 Rev3. Retrieved from https://store.arduino.cc/usa/mega-2560-r3. (Jan. 2021).

[49] Arjun Tambe, Akshay Nambi, and Sumukh Marathe. 2021. Is your smoke detector working properly? Robust fault tolerance approaches for smoke detectors. In *Proceedings of the 19th ACM MobiSys*. 310–322. DOI: https://doi.org/10.1145/3458864.3466869

[50] Guoming Tang, Zhen Ling, Fengyong Li, Daquan Tang, and Jiuyang Tang. 2017. Occupancy-aided energy disaggregation. *Computer Networks* 117 (2017), 42–51.

[51] Amal Tawakuli, Daniel Kaiser, and Thomas Engel. 2022. Transforming IoT data preprocessing: A holistic, normalized and distributed approach. In *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*. Retrieved from https://api.semanticscholar.org/CorpusID:256195144

[52] Amal Tawakuli, Daniel Kaiser, and Thomas Engel. 2023. Experience: Differentiating between isolated and sequence missing data. *ACM Journal of Data and Information Quality* 15, 2 (2023), 1–15.

[53] Y. Wang, A. Yang, Z. Li, X. Chen, P. Wang, and H. Yang. 2016. Blind drift calibration of sensor networks using sparse Bayesian learning. *IEEE Sensors Journal* 16, 16 (2016), 6249–6260. DOI: https://doi.org/10.1109/JSEN.2016.2582539

[54] John G. Webster. 1999. *The Measurement, Instrumentation, and Sensors: Handbook*. Springer Science & Business Media.

[55] Kamin Whitehouse and David Culler. 2002. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*. 59–67.

[56] W. Wu, X. Cheng, M. Ding, K. Xing, F. Liu, and P. Deng. 2007. Localized outlying and boundary data detection in sensor networks. *IEEE Transactions on Knowledge and Data Engineering* 19, 8 (2007), 1145–1157. DOI: https://doi.org/10.1109/TKDE.2007.1067

[57] Yun Xiang, Lan Bai, Ricardo Piedrahita, Robert P. Dick, Qin Lv, Michael Hannigan, and Li Shang. 2012. Collaborative calibration and sensor placement for mobile sensor networks. In *Proceedings of the 11th ACM IPSN*. 73–84. DOI: https://doi.org/10.1145/2185677.2185687

[58] Ke Yan and David Zhang. 2015. Improving the transfer ability of prediction models for electronic noses. *Sensors and Actuators B: Chemical* 220 (12 2015). DOI: https://doi.org/10.1016/j.snb.2015.05.060

Appendix

A Frequently Asked Questions (FAQs)

Q: Where does PIRMedic *detect and diagnose* faults?

A: PIRMedic performs detection and diagnosis on collected data at the edge device, and does not need any additional hardware. Edge devices are commonly present in sensor deployments and are responsible to aggregate data from multiple sensors and transmit it to the cloud. We use the edge to execute PIRMedic.

Q: Does PIRMedic require *cloud connectivity*?

A: PIRMedic performs all the computation on the edge and does not require cloud connectivity. As it detects failures on the edge and rejects incorrect data, this saves cloud-edge network bandwidth as well as ensures data from broken sensors do not reach the cloud, thereby enabling better data analytics at the cloud as well.

Q: Does PIRMedic detect faults at runtime? Is it instantaneous?

A: PIRMedic is used to detect failures by analyzing samples over a window of 256 − 1024 samples i.e., 5 − 20 seconds at a sampling rate of 50 Hz. Thus, it can be used to detect anomalies at runtime. In addition, PIRMedic is non-intrusive and does not involve disassembling or uninstalling the sensor. However, the typical use case is a post-analysis e.g., once in a day to find out failing sensors since edge devices are typically resource constrained.

Q: Does PIRMedic require special sensors?

A: PIRMedic requires a PIR sensor that exposes the A_{out} signal i.e., output of pyroelectric element. Currently, not all PIR sensor manufacturers expose A_{out} signal in their sensor. Note however that *every* PIR sensor contains the A_{out} signal, as it is a core part of the pyroelectric element physics.

Q: What can PIR sensor manufacturers takeaway from PIRMedic?

A: We use this work to make sensor manufacturers include the signal A_{out} in all their sensor designs as it would enable fault detection and diagnosis at the edge.

Q: Are the failures artificial?

A: It is a criticism that PIRMedic handles only artificially-induced failures. However, this is not true. The failures induced are those observed by IoT/Smart Building technicians and engineers who deploy sensors in our buildings in US and India. We recreate these failures on cheap sensors as it is not feasible to perform large-scale deployment with expensive sensors typically seen in modern buildings.

Q: Does PIRMedic work with high-grade PIR sensors?

A: While our work focused on low-cost, commodity, large-scale, cheap deployments of PIR sensors, the physics of sensing is identical in commercial-grade PIR sensors as well. We accessed a PIR sensor from our building ceiling and analyzed the A_{out} waveform by connecting it to an oscilloscope. In our preliminary observations, we saw that the shape of A_{out} is similar to that we saw in low-cost sensors. This similarity in the nature of A_{out} waveforms shows promise to that our analysis and techniques can be extended to commercial-grade PIR sensors as well. As each of these sensors costs around $200, we could not conduct a study on different failures due to their exorbitant cost.

Received 9 August 2024; revised 24 December 2024; accepted 5 May 2025