SILOS: An Intelligent Fault Detection Scheme for Solar Insecticidal Lamp IoTs with Improved Energy Efficiency

Xing Yang, *Graduate Student Member, IEEE,* Lei Shu, *Senior Member, IEEE,* Kailiang Li, Zhiqiang Huo, Sheng Shu, and Edmond Nurellari, *Member, IEEE*

***Abstract*—Solar Insecticidal Lamp Internet of Things (SIL-IoTs) nodes are susceptible to failures due to the harsh environment, burn-in, theft, and vandalism in the agricultural setting. Most state-of-the-art research mainly focuses on fault detection without considering hardware and communication performance in terms of potential fault modes, energy consumption, and network loads. This study presents a completely decentralized solution, namely a SIL-Oriented binary Sliding window-based fault self-detection scheme (SILOS), which can be performed on each SIL-IoTs node. The problem we are trying to answer is how to detect faults as accurately as possible while keeping the communication overhead, memory, and computational costs low. Specifically, we develop a fault dictionary concept to 1) model the faults of SIL-IoTs nodes, 2) construct the fault dictionary according to the characteristics of measurements, and 3) detect faults via the fault dictionary. In addition, a binary-based sliding window (BSW) fault self-detection approach is proposed to save detection costs and reduce the false alarm rate (with only 92B system caches). A series of experiments are performed to evaluate the performance of the proposed method. The result demonstrates that the BSW method can detect faults with an average accuracy of 99.14% with less than 1% energy consumption. By only sending the fault code, 71B data (i.e., data transmitting and forwarding) can be reduced, saving energy consumption, and decreasing network congestion.**

***Index Terms*—Fault Self-Detection, Solar Insecticidal Lamp Internet of Things, Arduino, Sliding Window, Fault Dictionary, Hardware Design**

This work was supported in part by the National Natural Science Foundation of China under Grant 62072248, in part by High-end Foreign Experts Recruitment Plan of the Ministry of Science and Technology China under Grant G2021145009L, in part by Jiangsu Modern Agricultural Machinery Equipment and Technology Demonstration and Promotion Project under Grant NJ2021-11, in part by International Cooperation Capacity Enhancement Plan of Nanjing Agricultural University under Grant 2021-PY-10, and in part by China Scholarship Council (CSC) under Grant 202106850023. *(Corresponding author: Lei Shu.)*

X. Yang is with the College of Engineering, Nanjing Agricultural University, Nanjing, 210031, China, and also with the School of Engineering, University of Lincoln, Lincoln, LN67TS, U.K. (e-mail: harryyangx@gmail.com).

L. Shu is with the College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, 210031, China, and also with the School of Engineering, University of Lincoln, Lincoln, LN67TS, U.K. (email: lei.shu@njau.edu.cn).

K. Li and S. Shu are with the School of Artificial Intelligence, Nanjing Agricultural University, Nanjing, 210095, China (email: Kailiang_li@njau.edu.cn and ShuSheng869@outlook.com).

Z. Huo is with the Institutional Research Information System, University College London, London, WC1E 6BT, U.K. (email: z.huo@ucl.ac.uk).

E. Nurellari is with the School of Engineering, University of Lincoln, Lincoln, LN6 7TS, U.K. email: (ENurellari@lincoln.ac.uk).

I. Introduction

The Solar Insecticidal Lamp Internet of Things (SIL-IoTs) is a key application in the smart Agricultural Internet of Things (Ag-IoTs). A typical SIL-IoTs system consists of the Solar Insecticidal Lamp (SIL) and Internet of Things (IoT) module [1]. By adopting IoT modules, SIL-IoTs nodes can 1) monitor meteorological information, their faulty status, and residual energy, 2) estimate their numbers of killed pests, and 3) transmit the above data to a server or edge node [2].

In most cases, SIL-IoTs nodes are deployed in the wild. Because of the harsh environment, SIL-IoTs nodes are vulnerable to unpredicted faults, such as line open-circuit (OC), circuit board damage, and lure lamp damage [3], [4]. Furthermore, the remote deployment location and large-scale network make SIL-IoT maintenance difficult. These issues have an impact on the long-term and dependable operation of SIL-IoTs. Furthermore, a lack of fault detection and isolation will result in serious consequences and safety accidents. For example, a faulty solar charge controller causes the battery to charge continuously and overheat, resulting in the battery exploding. As a result, effective and precise fault detection methods are required to ensure system performance and node life.

A. Motivations

- To the best of our knowledge, although fault detection on IoT has been extensively studied, fault detection of the SIL-IoTs node has rarely been studied systematically. The main practical problem that confronts us is what faults will occur based on SIL-IoTs' hardware modules.
- Since SIL-IoTs nodes are normally deployed in a large-scale scenario, the delay in transmitting data to a central server reduces the fault detection speed. Sending and forwarding data from nodes to a central server also consumes energy, impacting the remaining energy used for killing pests.
- Notably, SIL-IoTs nodes are usually deployed in the wild. It is hard for maintenance personnel to reach the faulty node to repair damaged modules. The accuracy of the detection model needs to be improved to reduce maintenance costs.
- In particular, the solar panel cannot harvest enough energy under continuously cloudy weather; thus, it is critical to transmit fewer data to reduce energy consumption.

Based on these, a fault self-detection scheme that performs fault detection on each node is applicable for SIL-IoTs. However, the SIL-IoTs node is resource-constrained, so the designed model cannot be too complex. It is critical for the fault detection scheme of SIL-IoTs to balance these two metrics. Consequently, the state-of-the-art research of SIL-IoTs should be studied to determine available measurements and describe faults based on the existing modules. In addition, developing an easy implementation, high accuracy, and lightweight fault self-detection scheme according to SIL-IoTs nodes' storing and computing capacity needs to be addressed.

B. Contributions

We aim to provide a completely decentralized solution that can be run on each SIL-IoTs node, which detects faults of SIL-IoTs nodes as accurately as possible while keeping the communication overhead, memory, and computational costs low. Our main contributions are as follows:

1) To the best of our knowledge, this is the first study that addresses the above issues (i.e., fault description and lightweight fault self-detection according to existing modules of SIL-IoTs nodes) by a systematic fault self-detection scheme (SILOS), including hardware design and a fault self-detection model.
2) We developed an improved SIL-IoTs node with enhanced functionality to measure faulty conditions and reduce faults of SIL-IoTs.
3) Based on the fault dictionary concept, faults of SIL-IoTs nodes are summarized and labeled systematically for the first time. Due to the limited energy and large-scale deployment of SIL-IoTs nodes, a concise report rather than all measured data is more suitable for fault detection. Therefore, faults of SIL-IoTs nodes are coded as one byte to reduce delay, energy consumption, and network traffic.
4) An energy/resource efficient implementation of our proposed binary-based sliding window fault self-detection algorithm is performed on the microcontroller (e.g., limited data storage and computing capacities, including 20MHz CPU speed, 32KB program memory size, and 2KB RAM size).

C. Organization

The rest of the paper is organized as follows: Section II describes the related work. Section III introduces the modules of the SIL-IoTs-V1.0, the new hardware design, and discusses the fault conditions. Section IV introduces the proposed fault self-detection approach. Section V introduces the experimental setup and the proposed method's parameter setting. Section VI discusses the performance of the proposed method. Section VII concludes this study. For easy presentation, we summarize the acronyms in this article in Table I.

II. Related work

A. The state-of-the-art of SIL-IoTs

The concept of SIL-IoTs was first proposed as an emerging Ag-IoTs in 2019 [5]. Four key issues were presented, and the state-of-the-art research is listed as follows.

- **Optimized deployment scheme of SIL-IoTs nodes with different constraint conditions:** In 2020, Yang *et al.* [1] proposed a node deployment strategy of SIL-IoTs based on partition structure. By utilizing the natural partition structure of farmland, the node deployment problem of SIL-IoTs was transformed into a secondary allocation problem, which was solved by a genetic algorithm. The proposed deployment strategy significantly reduces the number of nodes deployed while ensuring full coverage and use in complex agricultural environments. In 2021, weighted deployment [6] and hole-aware node deployment strategies [7] were proposed to improve the connectivity of SIL-IoTs and address the mixed-crop farmland problem.
- **Adaptive energy management strategy to optimize the energy utilization of SIL-IoTs:** Keeping the lamp on to attract pests and releasing high voltage pulse currents to kill pests occupy major energy consumption of a SIL-IoTs node, in which consumed energy comes from the residual energy of the battery and the energy harvesting of the solar panel. However, insufficient energy harvesting situations (e.g., long-term cloudy days and shadow shading of the solar panel) cause abnormal operation of SIL-IoTs. Besides, different discharge numbers of different SIL-IoTs nodes result in the inconsistent residual energy of nodes. In 2021, Guo *et al.* [8] proposed a two-hop energy consumption balanced routing algorithm named THECB to address these issues. The result indicates that THECB achieves lower energy consumption and better energy consumption balance of communication than baseline methods. In addition, a SILs management mechanism based on real-time information about pests and residual energy is presented to optimize the energy consumption of killing pests [9].
- **Pests outbreak area localization:** Migratory pests usually migrate to other areas and reproduce after a local area outbreak. Thus, the outbreak area of migratory pests shows the characteristics of a multi-area outbreak and rapid diffusion [5]. Therefore, counting the accurate number of killed pests is critical for monitoring outbreak areas and promptly taking corresponding measures. Currently, the SIL-IoTs-V1.0 count the number of killed pests by the high voltage discharge times of the metal mesh [10]. When more than one pest contacts the metal mesh simultaneously, only one killed pest is recorded because the metal mesh only discharges once. Furthermore, corpses of pests stick to the metal mesh, leading to continuous discharge. The counted discharge times are abnormal. Therefore, additional information is required to ensure the accuracy of counting the number of killed pests. For instance, Lopez *et al.* [11] proposed a pest monitoring system based on image sensors, which detects pests through an image recognition algorithm and uploads results by multi-hop data transmission.
- **Interference on data acquisition and transmission due to high voltage discharge:** Several works [2], [12], [13] point out that high voltage discharge of the metal mesh (as a kind of electromagnetic interference) impacts data

TABLE I
ACRONYMS FOUND IN THIS PAPER

Acronym	Description	Acronym	Description	Acronym	Description
SIL-IoTs	Solar insecticidal lamp IoTs	BSW	Binary-based sliding window	Ag-IoTs	Agricultural-IoTs
OC	Open-circuit	SIL-IoTs-V1.0	SIL-IoTs-version 1.0	SIL-IoTs-V2.0	SIL-IoTs-version 2.0
PCB	Printed circuit board	PWM	Pulse width modulation	V_p	Voltage pulse
A_f	Audio frequency	TH_v	Voltage pulse threshold	TH_S	Audio frequency threshold
K	number of killed pests	SC	Short circuit	R	Multi-characteristics matter
V^L	Lamp voltage	V^M	Metal mesh voltage	V^B	Battery voltage
WSNs	Wireless sensor networks	TP	True positive	FP	False positive
TN	True negative	FN	False negative	Acc	Accuracy
FAR	False alarm rate	MAR	Missing alarm rate	ROC	Receiver operating characteristic
AUC	Area under ROC curve	TPR	True positive rate	FPR	False positive rate
r	Pearson correlation coefficient	σ	Standard deviation	R^2	Degree of regression
a and b	Parameters of a linear regression	y_{est}	Estimated y	y_{obs}	Observed y

acquisition and transmission, which leads to 1) abnormal data and 2) failure of monitoring and predicting operation. To avoid the interference of discharge, Huang *et al.* [2] suggested that 25cm is a safe installation distance between the SIL and the wireless communication module.

Furthermore, the anti-theft and anti-destruction issue of SIL-IoTs node was discussed in [14]. With a new design, more information from different sensors (i.e., gated switch, voltage and current module, emergency power module, and acceleration sensor module) is obtained to protect SIL-IoTs nodes. Without an anti-theft and anti-destruction detection model, [14] only considers hardware design. Similarly, a survey paper studied the characteristics and challenges of fault diagnosis on SIL-IoTs without proposing a fault detection model and labeling faults of SIL-IoTs nodes [10].

In summary, there are growing appeals for studying SIL-IoTs. Recent research indicates that:

- The SIL-IoTs node is lowly redundant to reduce the deployment cost. The failure of one SIL-IoTs node may lead to communication interruption. Therefore, ensuring the stability of each node plays a vital role in the daily operation of a single node and the connectivity of the whole network.
- It is necessary to manage the energy of SIL-IoTs nodes through weather conditions prediction, residual energy estimation, and rhythm of pest to SIL-IoTs nodes. Based on this, the solar panel's energy conversion efficiency and the battery's working condition must be monitored.
- The condition that corpses of pests stick to the metal mesh should be detected timely for accurately counting the number of killed pests.
- Electromagnetic interference should be avoided to ensure data quality and fault detection accuracy.
- The information of additional sensors deployed to SIL-IoTs based on other research also contributes to the fault detection scheme.

All studies are based on the long-term and stable operation of SIL-IoTs nodes. However, no original paper related to the fault detection of SIL-IoTs nodes exists. Such neglect leads to the failures of SIL-IoTs nodes, impacting the application performance of related studies. Consequently, we study IoT equipment's fault detection strategies and methods to find a suitable solution for SIL-IoTs nodes.

B. Fault detection method

As shown in Table II, several typical scenarios involving fault detection strategies and their characteristics are investigated. Compared with industrial scenarios (e.g., [15]–[17]), SIL-IoTs nodes have attributes of 1) low sampling frequency, 2) limited computing and data storage capacities, 3) limited energy, and 4) large-scale, low density and outdoor deployment. These characteristics have the following impacts on fault detection performance:

- The increase in false alarm rate due to low timeliness and strong contingency data.
- The model's size, complexity, and energy consumption to be considered additionally.
- Unpredictable failure caused by the harsh environment.
- An increase in detection delay due to long-distance data transmission.
- Bandwidth occupation, data privacy and energy consumption due to a large amount of data transmission.

Therefore, it is critical for SIL-IoTs to select a suitable fault detection strategy. According to the implemented scheme of each fault detection method, fault detection methods are usually divided into two different types: quantitative strategy and qualitative strategy [10].

1) Quantitative strategy: A quantitative strategy usually performs the fault detection scheme in the sink node or a server, adopting global information to determine faulty degrees. Data-driven methods and model-based methods are generally adopted in the quantitative strategy.

By adopting numerous data and well-defined model structures to train a fault detection model, data-driven methods have high detection accuracy [20], [21], [30]. However, data-driven methods normally have high computing power and energy

TABLE II
TYPICAL FAULT DETECTION SCENARIOS, THEIR CHARACTERISTICS, AND FAULT DETECTION STRATEGIES

Paper	Scenario	Implement	Network	Sampling frequency	Energy		Deployment			Fault detection		
					Limited	Harvesting	Scale	Inter-node distance	Scene	Strategy	Computing capacity	Energy consumption
[15]	Rotating machinery	Server	N/A	12kHz	N	N	Low	N/A	Indoor	Quantitative	Not considered	Not considered
[16]	Industrial robot	Server	ZigBee	123Hz	N	N	Low	N/A	Indoor	Quantitative	Not considered	Not considered
[18]	Wearable device	Server	N/A	N/A	N/A	N/A	Low	N/A	Indoor	Quantitative	Not considered	Not considered
[19]	IoT farm system	Server	N/A	N/A	N/A	N/A	N/A	N/A	Outdoor	Quantitative	Not considered	Not Considered
[20], [21]	WSNs	Server	N/A	0.2-2Hz	N/A	N/A	N/A	N/A	Both	Quantitative	Not considered	Not Considered
[22]	Petrochemical plant	Server	N/A	1/60Hz	N/A	N/A	N/A	N/A	Indoor	Quantitative	Considered	Not Considered
[23]	Photovoltaic series	Compact-RIO	N/A	20kHz	N	N	Low	N/A	Outdoor	Quantitative	Considered	Not Considered
[24]	Environment surveillance sensor	Cluster head	N/A	1/30Hz	N/A	N/A	Medium	N/A	Indoor	Qualitative	Not considered	Not considered
[25]	Surveillance device for island invasion detection	Arduino	ZigBee & GPRS	10Hz	Y	Y	Medium	1.01m	Outdoor	Qualitative	Considered	Mentioned
[26]	Surveillance device for forest	MSP430	ZigBee	N/A	N/A	N/A	Large	N/A	Outdoor	Qualitative	Considered	Not Mentioned
[17]	Three-phase voltage-source inverter	DSP	N/A	50kHz	N	N	Low	N/A	Indoor	Qualitative	Considered	Not Considered
[27]	Photovoltaic power generation systems	PIC Single-Chip	ZigBee	N/A	Y	N/A	Low	N/A	Outdoor	Qualitative	Considered	Mentioned
[28]	Plant-Wide Process Monitoring	N/A	N/A	N/A	N	N	Medium	N/A	Indoor	Qualitative	Not Considered	Not Considered
[29]	Air-handling units	N/A	N/A	N/A	N	N	Low	N/A	Outdoor	Qualitative	Not Considered	Not Considered
Our	SIL-IoTs	Arduino	ZigBee	0.2Hz	Y	Y	Large	[60m, 120m]	Outdoor	Qualitative	Considered	Considered

consumption requirements. Although advanced deep learning concepts (e.g., federated learning and lightweight convolutional neural networks [23]) have been widely promoted, it is difficult for the microcontroller to apply data-driven methods.

Model-based methods detect faults by applying a dynamic process to estimate signals and parameters [31], e.g., gray box [32] and state observers [33], [34]. The faulty current or voltage sensor can be isolated by comparing the residual and the predefined threshold [35], [36]. Besides, Multi-state observers contribute to the improvement of detection accuracy. For instance, Yu *et al.* [37] integrated three independent observers to monitor the state of the current sensors under normal operation. Model-based methods cost less computing capacity and energy consumption than data-driven methods in the microcontroller of SIL-IoTs nodes. However, model-based approaches have a slow time response [30].

Although a centralized strategy usually has a great detection accuracy and does not need to consider the constraints of computing capacity and energy consumption, it has to transmit a large amount of data to the central server, which results in the occupation of bandwidth, the loss of local information, and the increase of energy consumption of data transmission.

2) Qualitative strategy: IoT nodes usually have limited energy and computing capacity, which indicates a trade-off between the accuracy and lightweight of the fault detection model according to the characteristics of IoT nodes. By deploying the detection scheme to IoT nodes, the qualitative strategy can overcome the disadvantages of the quantitative strategy.

The qualitative strategy-based prior knowledge method can be adopted to reduce the modeling cost greatly, and calculation cost considering the computing and data storage capacities of microcontroller [30]. For instance, Sun *et al.* [38] proposed a structural graph and clustering-based machine fault diagnosis scheme that assigns categorical weights by the distance between expressed and expressing data. Yan *et al.* [17] proposed

a fuzzy logic-based method to detect multiple OC faults of the three-phase voltage-source inverter. Liu *et al.* [29] detect three types of fault severities (named small, medium, and large faults) by the Bayesian inference, which only uses a small amount of measurement data for estimation. These schemes can detect faults accurately. However, they are only performed on simulation rather than real IoT nodes.

A self-learning dictionary and fault tree analysis-based fault diagnosis method was designed for detecting the faults of outdoor and energy-limited IoT nodes and performed in real-world test-beds [39]. More specifically, a fault is detected if the offset exceeds the rating of $\pm 20\%$. However, detecting faults only by a single data point increases the false alarm rate due to data fluctuation. To reduce the false alarm rate, Liu *et al.* [26] adopted a window-based scheme for caching the latest parameter values. Then, the cumulative sum charts of the current data sequence are calculated to localize the change-point (as faulty sensor evidence). The trouble-shooting root causes of 330 sensor nodes deployed in the forest can be detected based on local evidence. The false detection rate can also be reduced through the joint voting decision of multiple nodes. For instance, a sensor-level and system-level anomaly detection method TrusDet was proposed to prevent unauthorized invasion of outdoor IoT nodes [25]. TrusDet adopted the following steps to detect an intruder accurately: 1) selecting past slots as a sliding window to obtain a fused result, 2) obtaining the binary output according to the chosen threshold, and 3) determining faulty conditions via voting of nearby nodes. This scheme does not take advantage of the correlation between measurements within individual IoT nodes to improve the efficiency of anomaly detection. In addition, transmitting to the sink or cluster head leads to additional data transmission and energy consumption. The failure of multiple neighbor nodes also affects the fault detection results in the current cluster.

C. Summary

SIL-IoTs nodes have the characteristics of 1) outdoor and large-scale deployment and 2) limitations on data storage, computing resource, and energy supply. From Table III, the qualitative strategy is more suitable for fault detection of SIL-IoTs nodes than the quantitative strategy. The advantages of existing qualitative based prior knowledge methods for outdoor and energy-limited IoT equipment are:

- By local decision-making, prior knowledge methods reduce the network congestion and energy consumption of data transmission.
- It provides real-time detecting results.
- Prior knowledge methods avoid the loss of information during transmission to improve detection accuracy.

Nevertheless, these works do not consider the energy consumption of fault detection schemes and how it will impact the whole network's energy consumption. Besides, existing prior knowledge methods have several limitations considering the characteristics of SIL-IoTs, as mentioned in Section II-A. Generally, IoT nodes have limited computing and data storage capacities, which indicates that the fault detection scheme should not be complex. However, in many cases, it is difficult to determine the root cause based on a single-time measurement of one sensor. Besides, problematic sensors usually cannot detect faults. A simple scheme results in a high false alarm rate and a low accuracy.

TABLE III
COMPARISON OF QUANTITATIVE AND QUALITATIVE FAULT DETECTION METHODS

Strategy	Quantitative		Qualitative
Method	Data-driven	Model-based	Prior knowledge
Accuracy	↑	–	↑
Robustness	↑	–	↓
Timeliness	↓	–	↑
Computing cost	↑	–	↓
Demand of rule	↓	↑	↑
Energy cost	↑	–	↓

↑,–, and ↓ denote high, medium, and low, respectively.

Particularly, considering that additional modules are added for SIL-IoTs nodes to achieve intelligent operations, the failure of fault detection methods caused by problematic sensors can be reduced via the correlation of multiple elements of a single node.

III. HARDWARE AND FAULT DESCRIPTION

According to Section II, there is no previously published paper on the fault detection of SIL-IoTs. Therefore, it is necessary to introduce the hardware module and workflow of a SIL-IoTs node. In this section, we introduce the modules of the SIL-IoTs-V1.0 and our new design to answer the following questions:

- What hardware modules does a SIL-IoTs-V1.0 node have?
- How are these modules connected?
- What are the daily tasks of a fault-free SIL-IoTs node?
- What can be improved in hardware design?
- What faulty conditions usually occur?

A. Hardware modules of SIL-IoTs-V1.0

As shown in Fig. 1, a SIL-IoTs-V1.0 node comprises SIL and IoT modules, which perform different tasks. During the daytime, SIL-IoTs-V1.0 nodes harvest energy from the solar panel and store energy by the battery. During the night time, the SIL kills pests using the harvested energy. If rain is detected, the lure lamp and metal mesh will be turned off by the command of Arduino. Finally, the sensing data and the number of killed pests are transmitted to the cloud or edge node by ZigBee.

As shown in Fig. 1 and introduced in [7], the SIL-IoTs-V1.0 usually contains the following modules:

- A solar panel is adopted for harvesting energy.
- A battery is adopted for supplying energy and storing energy.
- A lure lamp is adopted for attracting phototactic pests.

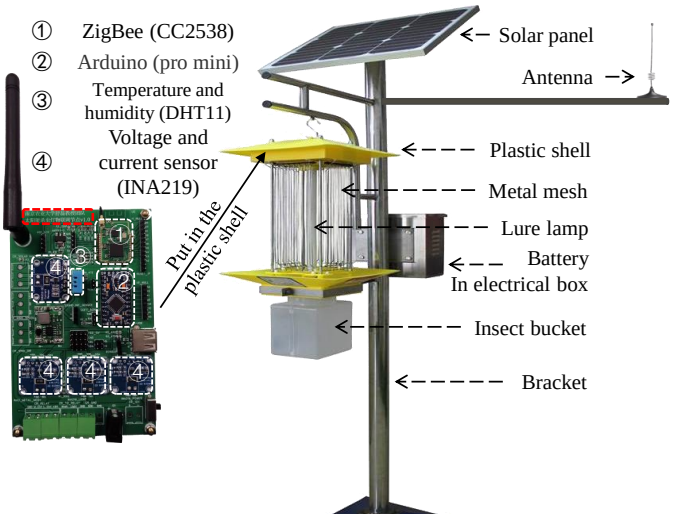


Fig. 1. Prototype and Printed circuit board (PCB) design of a SIL-IoTs node (introduced in [7], named SIL-IoTs-V1.0 for a comparison with our new design), where PCB, solar charge controller, IoT module, and ZigBee are deployed in the plastic shell. The content in the red dashed box indicates that "Nanjing Agricultural University, Team of Professor Lei Shu (SIL-IoTs-V1.0 node)".

- An ignition coil is adopted for converting the 12V output by the battery into a high voltage between 2150V and 6000V.
- A metal mesh is adopted for killing pests by releasing high voltage pulse discharge while pests contact with it, which leads to the change in the form of voltage fluctuation and noise.
- A voltage divider module is adopted for counting the number of killed pests by catching the voltage variation as shown in Fig. 5.
- An electronic transistor switch is used to turn on the lure lamp and metal mesh during the daytime and turn them off at night.
- A PCB is used to send a control command to actuators and gather the data acquired by the node.
- A rain fall detector is applied for detecting the resistance decrease due to raindrops. Then, the command (turn off the lure lamp and the metal mesh) is triggered by the change of voltage value.
- A plastic shell is used to fix the metal mesh, lure lamp, rain fall detector, and store the control circuit, ignition coil, solar charge controller, PCB, IoT module, and wireless communication module.
- A stainless steel bracket is adopted for hanging the SIL-IoTs node at a suitable height.
- An pest bucket is adopted for storing killed pests.
- A Pulse width modulation (PWM) based solar charge controller is adopted to improve energy harvesting efficiency and protect the battery by shutting off the supply once the real-time battery voltage value is lower than 10.7V.
- An electrical box is applied for storing the battery.
- A wireless communication module (ZigBee) is adopted for data forwarding and transmitting.
- The inter-node distance of SIL-IoTs nodes is between 40m to 70m [1]. Therefore, sensors can obtain micro-scale environmental conditions, e.g., temperature, humidity, wind direction, and wind speed. Besides, four voltage and current sensors are used to manage the energy of the SIL-IoTs node and monitor the battery, lure lamp, metal mesh, and solar panel. More specifically, the voltage and current values of the battery are used to evaluate the remaining energy. Meanwhile, voltage and current values of the solar panel are used to estimate the harvested energy. The faults of the battery, lure lamp, metal mesh, and solar panel can be estimated by voltage and current values [10].

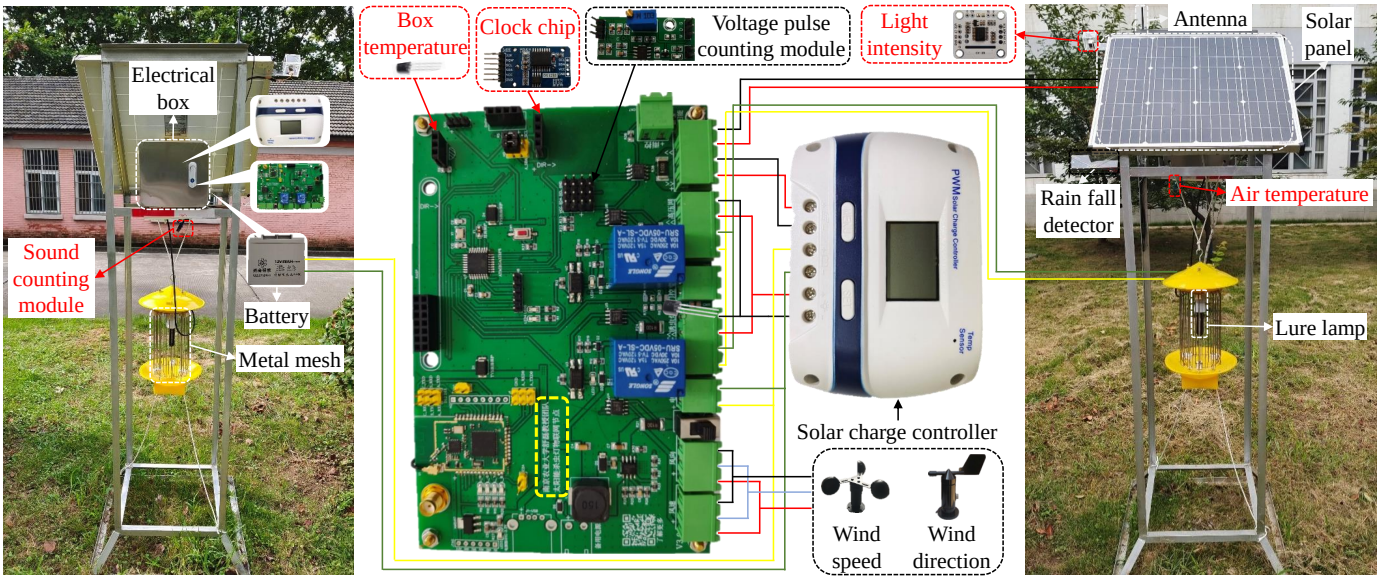
In general, the SIL-IoTs-V1.0 has the following drawbacks:

- High voltage discharge impacts data acquisition and transmission (as a kind of electromagnetic interference), which results in abnormal measurement [2], [12]. Thus, high voltage discharge impacts the accuracy of fault detection.
- Some faults cannot be detected due to the incomplete fault detection mechanism, e.g., the heating problem of battery and reduction of photoelectric conversion [10]. Furthermore, a single-time measurement of one sensor may lead to a false alarm, which reduces the reliability of the reported faulty information.
- The corpses of pests may stick to the metal mesh and cause continuous discharge, which will impact the insecticidal performance of the SIL-IoTs node [13]. The maintenance of the SIL-IoTs-V1.0 relies on a periodic manual inspection rather than remote and intelligent detection via acquired data.
- Turning on and off the SIL only via the electronic transformer switch may lead to energy waste and depletion. When the remaining energy is low, the lure lamp and metal mesh should be turned off to avoid an unexpected shutdown caused by a depleted battery, reducing system failures and prolonging battery life.

B. New hardware design

To address the drawbacks of the SIL-IoTs-V1.0 and four key issues presented by [5], we improve the hardware design, namely SIL-IoTs-V2.0, as shown in Fig. 2 (structure and circuit connection of the SIL-IoTs-V2.0) and Fig. 3 (logical connection of the SIL-IoTs-V2.0). The working flowchart of the proposed SIL-IoTs system is shown in Fig. 4, where the red part is the new or optimized function. The improvements are listed as follows:

1) Reducing electromagnetic interference: In SIL-IoTs-V1.0, IoT modules, the control circuit, and the ignition coil are usually placed in the plastic shell. The short distance between these modules and the metal mesh results in electromagnetic interference in data acquisition and transmission [12]. We only place the ignition coil in the plastic shell and other modules in the electrical box to reduce electromagnetic interference. There is a distance of more than 25cm between the electrical box and



(a) Behind of a SIL-IoTs node (b) Circuit connection of a SIL-IoTs node (c) Front of a SIL-IoTs node

Fig. 2. New design of a SIL-IoTs node. (a) and (c) is the new structure of SIL-IoTs-V2.0. (b) is the circuit connection of SIL-IoTs-V2.0. Red words represent new modules. Red and yellow lines denote the live line. Green and Black lines denote the neutral line. Blue lines denote the data acquisition of the wind speed and wind direction sensors. The content in the yellow dashed box indicates that "This PCB of SIL-IoTs node is designed by the research team of professor Lei Shu from Nanjing Agricultural University, China".

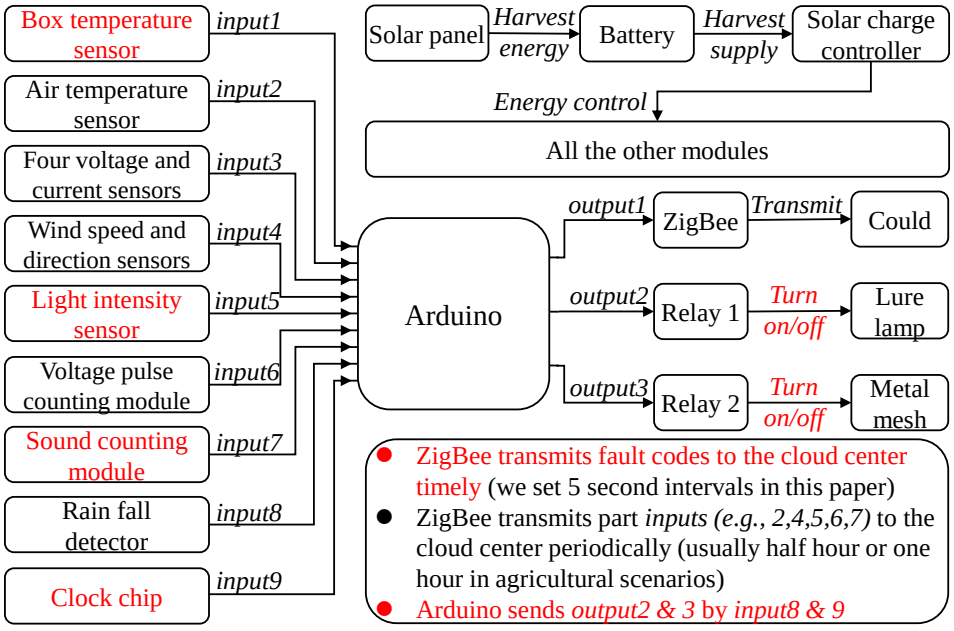


Fig. 3. The logical connection of SIL-IoTs-V2.0's hardware modules. Nine inputs are acquired to Arduino, and parts are transmitted to the cloud center. The red content is the improvements of the new design.

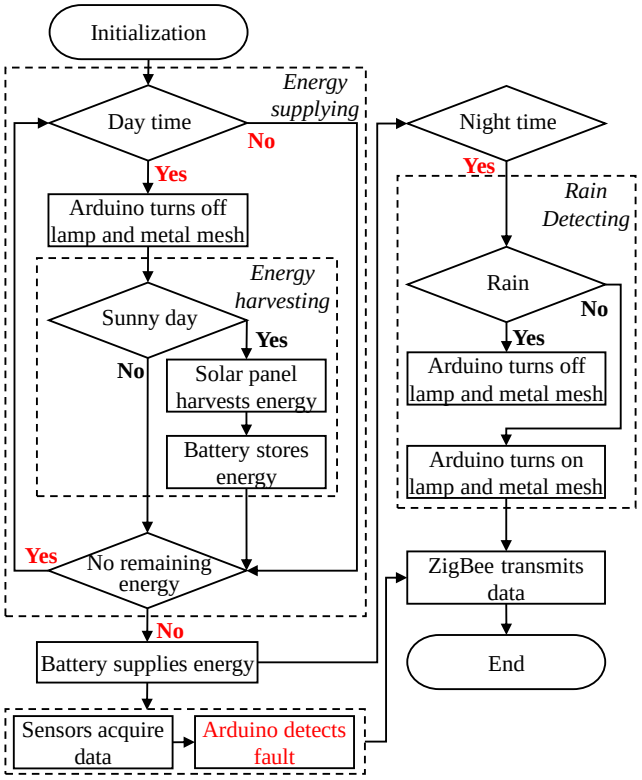


Fig. 4. The working flowchart of SIL-IoTs-V2.0 nodes' system, where the working time of SIL-IoTs nodes can be determined by the clock chip. The remaining energy can be measured by voltage and current sensors. Faults are detected via SIL-IoT nodes directly rather than transmitting to the cloud center.

the metal mesh to reduce electromagnetic interference [2]. In addition, we added multiple microvias on the circuit board to minimize electromagnetic interference.

2) Improving the fault detection mechanism: To detect the ignored faults [10], we add a temperature sensor (DS18B20, placed in the electrical box) to detect the heating problem of the battery. Besides, a high correlation exists between light intensity and the harvested power of the solar panel [40]. Thus, we monitor the solar panel according to the correlation.

3) Counting the number of killed pest accurately and detecting the corpses of pests sticking at the metal mesh: The SIL releases high voltage pulse discharge while insects contact the metal mesh, which leads to the change in the form of voltage fluctuation. Therefore, voltage pulse V_p and audio frequency A_f measurements accurately contribute to counting the number of killed pests in a period. In the SIL-IoTs-V1.0, a voltage divider module is used to catch the voltage variation caused by a voltage pulse. Because data acquisition and transmission also lead to voltage fluctuation, a threshold is predefined to avoid these situations by numerous experiments. The voltage pulse is cumulative once the voltage fluctuation exceeds the threshold Th_v. Similarly, a microphone module that converts sound signals into electrical signals counts the signals exceeding its predefined threshold Th_s. The accurate number of killed pests (represented as K) can be obtained by both modules, which contribute to locating pest outbreak

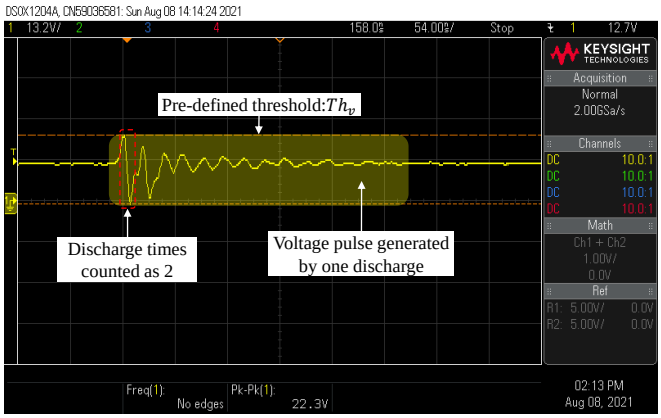


Fig. 5. Voltage fluctuation caused by one discharge. The voltage fluctuation is acquired by an oscilloscope, and the sampling frequency is 2.0 GSa/s.

areas, analyzing the activity rhythm of insects, and estimating the energy consumption of killing pests.

The corpses of pests stuck to the metal mesh will lead to continuous discharge. The voltage fluctuation caused by continuous discharge is low; thus, the number of V_p exceeding Th_v is nearly zero. However, sharp sounds will be recorded continuously. Therefore, the corpses of pests sticking to the metal mesh can be detected via the difference between the number of V_p exceeding Th_v and the number of A_f exceeding Th_s.

4) Adaptive energy management by intelligent turning on and off the SIL: The electronic transistor switch is replaced by a clock chip to turn on and off the SIL. In this way, the time of turning on and off the SIL can be adjusted according to the rhythm of the pests. Besides, the SIL will be turned off when the SIL-IoTs node's remaining energy is low or hard to harvest enough energy tomorrow due to the weather condition.

C. Fault condition

Based on hardware modules and the working flowchart of SIL-IoTs nodes. The faulty conditions of SIL-IoT nodes can be summarized as follows:

- **Circuit failure:** The circuit plays a vital role in controlling all modules of a SIL-IoTs node. Nevertheless, the circuit may fail due to aging, wear and tear, harsh environment, overuse, and defects, in which OC and short circuit (SC) faults are usually used to simulate these situations [41]. The OC fault occurs mainly due to the crack of the bond wires and gate driver fault, which considerably reduces the system performance considerably [42]. As the most serious fault, short-circuit (SC) fault leads to a huge current flowing through the circuit and then induces irreversible damage to modules [43].
- **Lamp damage:** Due to the harsh environment and limitation of a lifetime, the lure lamp is the component with the most frequent failure. Lamp damage caused by vandalism, short-circuit, and aging fails to attract pests of SIL-IoTs nodes, as shown in Fig. 6. In addition, repeated changes of the rain fall detector value lead to frequent switching of the lamp, damaging the lure lamp.

Fault-free lure lamp Damaged lure lamp

Fig. 6. The working conditions of the fault-free lure lamp and the damaged lure lamp at 11 p.m, where night mode is used to make photos bright at night.

Example A Example B

Fig. 7. Two examples of pests glue to metal mesh, which leads to continuous discharge, abnormal discharge, and impacts on estimating the number of killed pests. The red dashed box represents where the pest corpses are glued to the metal mesh.

- **Continuous discharge of metal mesh:** When pests contact both positive and negative metal meshes, a high voltage pulse will be discharged until the pest is shot down. However, some pests may be glued to the metal mesh, leading to continuous discharge unless burnt dry, as shown in Fig. 7. Furthermore, other conductive objects stuck to metal meshes will also cause continuous discharge, e.g., sensors and rainwater. Such a fault impacts the discharge voltage of metal meshes and even causes the SIL-IoTs node to fail to kill pests properly. When the metal mesh is not properly discharged, the pests attracted by the lights will continue to gather, inevitably leading to pest outbreaks and crop losses.
- **Damage of solar panel:** Due to line aging, dust coverage, glass panel fragmentation, and foreign matter shading, the solar panel's photoelectric conversion rate are bound to decline, resulting in a decline in energy harvesting efficiency. Moreover, the hot spot formed by long-term shading cannot be ignored. The hot spot may increase local temperature, local cell burning, solder joint melting, and other failures in the solar panel.
- **Burn-in of battery:** Similar to the solar panel, burn-in results in battery capacity degradation. The electrical box is usually designed as a semi-enclosed structure to prevent theft, rain, insects, and other factors, which leads to poor ventilation performance and further increases the temperature in the box. The continuous high temperature will accelerate the aging of the battery [44]. In addition, the battery is undercharged before the next discharge due to the uncertainty of weather conditions [45], which affects the other modules.
- **System failure:** Due to various causes, control systems' temporary failure is often encountered [46]. These failures can cause problems such as incorrectly formatted data, delays in switching lights on and off, and failure to store data. For instance, if the clock chip is restarted due to system failure, the system time of the SIL-IoTs node cannot correspond to the actual time, resulting in abnormal light-on time.
  Notably, based on [2], [12], high voltage discharge of metal mesh (a kind of electromagnetic interference) impacts data acquisition and transmission. As a transient fault, electromagnetic interference will disappear soon. Thus, system restart or maintenance is not required when electromagnetic interference occurs. However, low-quality data and missing data caused by electromagnetic interference increase the false alarm rate of fault detection methods.

IV. Proposed Method

In this section, the fault dictionary concept for SIL-IoTs nodes is introduced in three phases. Then, faults of a SIL-IoTs node are listed. The characteristics of these faults are described by the matter-element theory. At last, the BSW method is introduced as the fault detection scheme.

A. Fault dictionary concept

The fault dictionary concept is a typical simulation before test method [47]. The fault dictionary concept consists of three phases.

- **Fault modeling:** Fault characteristics are stored in the fault dictionary for future fault detection.
- **Dictionary constructing:** All the potential faults and fault-free cases are simulated. Then, the output characteristics are stored and classified in the dictionary.
- **Fault detecting:** According to the preset criteria, the measurement characteristics are compared with the predefined fault dictionary. Detection accuracy, false alarm rate, and missed fault rate can be used to evaluate the fault dictionary.

B. Fault modeling

This paper aims to perform a fault self-detection scheme via resource-constrained IoT nodes. In this way, accurate and

timely messages of faults from the scheme are provided for maintenance personnel to remove the fault, which contributes to reducing the continuous damage caused by the fault. Therefore, faults of the sensors used to monitor modules of SIL-IoTs nodes (e.g., voltage and current sensors) and faults that are difficult to detect in a short time through a single node (e.g., slight changes of data due to data drift) are out of the scope of this paper. According to faulty conditions mentioned in Section III-C, faults that can be detected by self-information of the SIL-IoTs node are listed in Table IV, where the fault-free condition is labeled as F0. When a fault is detected, the hexadecimal fault code listed in Table IV will be sent to the server through ZigBee.

A dictionary is usually composed of several key-value pairs. In this paper, the keys represent the fault labels, and the values represent the observations that are used to detect the corresponding faults, where values in all key-value pairs can be obtained by various experiments. According to Fig. 2, the modules to be continuously monitored in a SIL-IoTs node include the lure lamp, metal mesh, battery, solar panel, rain fall detector, and sensors for measuring air temperature and humidity (DHT11), electrical box temperature (DS18B20), light intensity, wind direction, and wind speed. The faults of the lure lamp, metal mesh, battery, and solar panel are monitored by four voltage and current sensors, respectively, while the other modules are evaluated by the status of their own measurements.

The fault that may occur in all of these modules is the OC fault. When the OC fault occurs, the current values of the lamp, metal mesh, battery, and solar panel are significantly reduced. Therefore, the OC fault of the corresponding module is used as the key, and the current under that fault is used as the value, thus forming a key-value pair. It is notable that lamp damage due to burn-in or vandalism leads to different current readings compared with the OC fault of the lure lamp. Thus, another key-value pair is required for the lamp damage fault. For sensors, they will lose energy supply due to the OC fault and cause measurement anomalies. Similarly, a key-value pair is constructed by using the OC fault of the corresponding sensor as the key, and the observation under that fault as the value. In particular, the rain fall detector is always disconnected. The rain fall detector is connected when rain drops onto it, which changes its voltage value. Therefore, it is not necessary to detect the OC fault of the rain fall detector. However, because the rain fall detector determines the on/off of SIL, it is essential to detect misjudged rainfall due to the SC fault of the rain fall detector. Since the difference in voltage changes due to the SC fault and rain, a key-value pair can be constructed by using the SC fault as the key and the voltage under that fault as the value.

Other faults listed in Table IV are caused by mismatches between multiple measurements. For example, the metal mesh continuous discharge can be detected by the difference between voltage pulse count and sound count. Therefore, a key-value pair is constructed using the metal mesh continuous discharge fault as the key and voltage pulse count and sound count as values. Similarly, key-value pairs of F8, F10, and F13 can be obtained by the corresponding measurements. In addition, the clock chip fault causes the lure lamp and metal mesh to be turned on during the day. Thus, this fault can be estimated by comparing current values of the lure lamp and metal mesh (higher when they have been turned on) and the light intensity sensor's values (higher during the day).

TABLE IV
LABELING OF FAULTS

Fault status	Label	Code
Fault-free	F0	0x00000000
Lamp OC	F1	0x00010001
Lamp damage	F2	0x00010010
Metal mesh OC	F3	0x00100001
Metal mesh continuous discharge	F4	0x00100010
Solar panel OC	F5	0x00110001
Battery OC	F6	0x01000001
Light intensity sensor OC	F7	0x01010001
Mismatch between light intensity sensor's value and solar panel's value	F8	0x01010010
Rain fall detector SC	F9	0x01100001
Mismatch between rain fall detector's value and light intensity sensor's value	F10	0x01100010
DHT11 OC	F11	0x01110001
DS18B20 OC	F12	0x10000001
Mismatch between DHT11's value and DS18B20's value	F13	0x10000010
Wind direction sensor fault	F14	0x10010001
Wind speed sensor fault	F15	0x10100001
Clock chip fault	F16	0x11000001

C. Dictionary constructing

We obtain various key-value pairs for different faults based on the previous section. The next step is to find a tool that can describe these key-value pairs in a normalized way. By listing the characteristics of measurements as a vector, the matter-element theory can be applied to construct a fault dictionary [27]. For a matter N with a characteristic c the corresponding value of v, the matter-element can be explained as:

$$R = (N, c, v) \tag{1}$$

Where R denotes the characteristic set of N. For instance, R = (DS18B20, T^1_{F0}, (0,60)) can signify that the DS18B20 is fault-free (denoted as T^1_{F0}) in the SIL-IoTs scenario when the reading ranges from 0 to 60 °C. For multi-characteristics matter, the matter-element is expressed as:

$$R = (N, c, v) = \begin{bmatrix} N, c_1, v_1 \\ c_2, v_2 \\ .. \\ c_n, v_n \end{bmatrix} \tag{2}$$

Where C=(c_1,c_2,...,c_n), V=(v_1,v_2,...,v_n) denote the characteristic vector and its value, respectively. For instance, R =

(DHT11, T^0_{F11}, 0; H_F11, 0) denote the F11 conditions of DHT11 if T^0_{F11} or H_F11 equals to 0.

Based on the matter-element theory, all modules of the SIL-IoTs node can be represented as several matter-element vectors.

D. Fault detecting

Based on the fault dictionary approach, potential SIL-IoTs faults are listed. Then, fault events and phenomena are constructed via the matter-element theory. The final step is to detect the SIL-IoTs' faults by the proposed BSW method. Because transient faults (e.g., electromagnetic interference) occur and disappear suddenly, the proposed fault self-detection scheme does not consider them. However, they impact the classification of faults labeled in this paper. Thus, we proposed the BSW method to reduce the false alarm rate.

1) Binary-based sliding window: A sliding window is a sub-list of data selected from a fixed data length under a certain step size [48]. As shown in Fig. 8 (a), we adopt one byte (8bits, each bit can store a binary number) to store the last eight data states for each measurement. In this way, the measurements in a short period can be stored for time-series analysis to reduce the false alarm rate. As represented in 3, if three binary zeros (abnormal states) appear in a sliding window, the component is defined as the faulty condition. f, x, and R_{F0} denote the state of the measurement, the current data point, and the matter–element vector's fault-free characteristic.

In other words, the component will be determined to be a fault condition if the detection threshold is suppressed three times out of eight consecutive data points. Then, the fault condition is transmitted as a fault code (one byte) via wireless sensor networks. In this way, at least three data points are required to detect faults of SIL-IoTs nodes.

$$f = \begin{cases} 0, & x \notin R_{F0} \\ 1, & x \in R_{F0} \end{cases} \tag{3}$$

2) Reducing detection delay by multi-variables: We adopt the correlation between component measurements for fault self-detection to reduce fault detection delay. As shown in Fig. 8 (b), we evaluate multi-variables for detecting F6. In this way, $\frac{1}{3}$ and $\frac{2}{3}$ detection delays are reduced by two and three variables, respectively.

3) Mismatch analysis between two modules: The mismatch fault cannot be detected via only one kind of measurement. For instance, as shown in Fig. 9, the voltage and current of the solar panel are not consistent with the light intensity value, which is drawn as the red area. In this situation, a threshold test cannot determine this type of fault via a simple threshold detection of voltage or current values. The high correlations of these three values can be used to detect this type of fault [49]. The following steps are adopted to detect this example: 1) establishing the equation and determining the coefficients by linear regression method; 2) performing the threshold judgment by the known equation after receiving the newest data point; 3) updating binary zeros in the sliding window if the threshold judgment rule is met; 4) sending the fault code "F8" when there are three zeros in the sliding window.

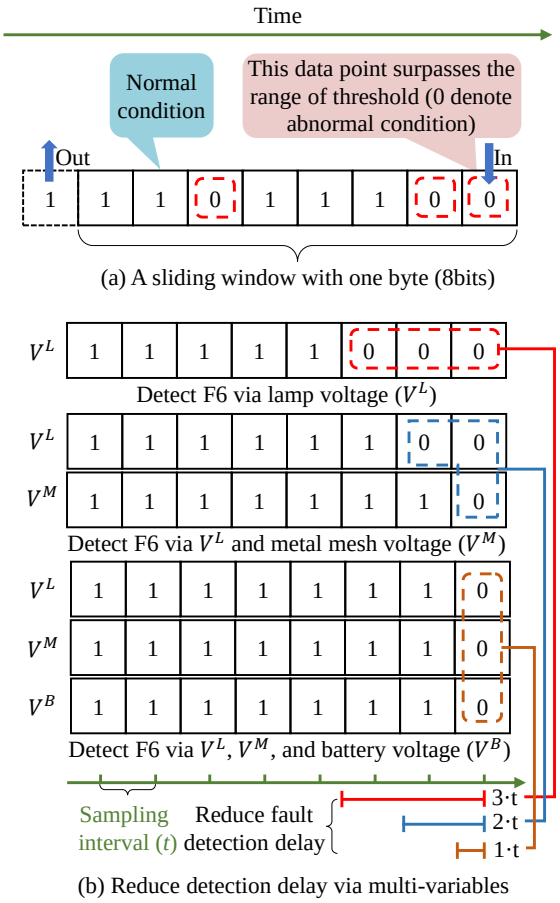


Fig. 8. (a) The binary data-based states of the recent 8 data points are stored in a sliding window with one byte (8bits). (b) An example of reducing detection delay by multi-variables, V^L, V^M, and V^B denote the voltage of the lamp, metal mesh, and battery. Detecting fault via single variable costs at least three sampling intervals. When detecting fault via two or three variables, they only cost at least two and one sampling interval, respectively.

In addition, the difference between DHT11 (used to acquire air temperature and humidity) and DS18B20 (used to acquire electrical box temperature) can be used to calculate the heating in the electrical box to assist in detecting the working state of the battery. In particular, there will be a low light intensity value if it rains, and the rain fall detector's value should fall below 900. Thus, there must be a contradiction between both when the light intensity value is high and the rain fall detector's value is under 900.

E. Summary

Based on these concepts, the proposed fault-detection scheme's flowchart is shown in Fig. 10. In the first phase, the prior knowledge is modeled by fault modeling. If the data format is wrong, a new data point will be recollected in the second phase. Otherwise, threshold detection will update the binary value into sliding windows. The total number of binary "0" will be counted. If the conditions are not met, the above operations will be repeated after receiving a new data point.

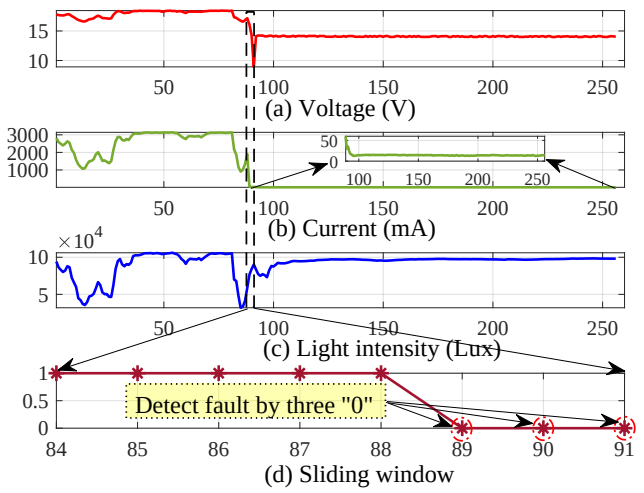
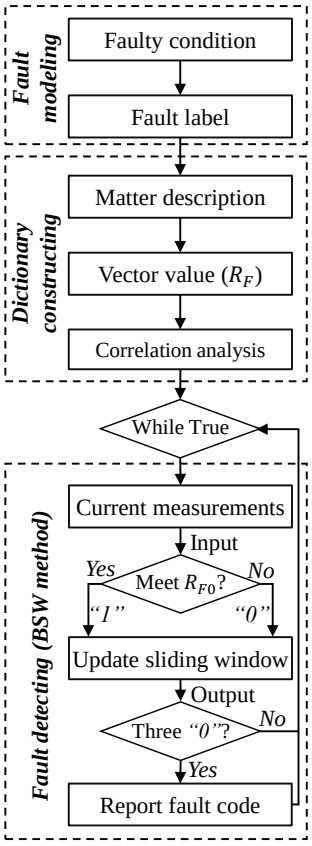


Fig. 9. An example of detecting F8 via consistency between voltage and current of the solar panel and the light intensity value, where voltage, current, and light intensity values do not exceed the preset thresholds.



- List fault that can be detected by single node in Section III-C;
- Report fault only via known hexadecimal code (1 byte) listed in Table III.
- Set threshold of each fault via matter-element theory in Section IV-C and experimental setup in Section V-A;
- Analyze correlation between elements to detect mismatch fault in Section IV-D and Figure 8.
- R_{F0} denotes characteristic set of fault-free condition;
- Store history faulty conditions via 16 sliding windows (16 bytes) that corresponding to F1-F16;
- Update more than one zero in a sliding window if multi-variables surpass R_{Fi};
- Detect faults in the order of F1 to F16;
- Report a fault code if a fault is detected, end this cycle and enter the next cycle.

Fig. 10. Flowchart of proposed fault self-detection scheme. Firstly, 17 fault conditions are labeled in the fault modeling phase. Secondly, matter descriptions (detecting thresholds) of all fault labels and correlations between elements are proposed and analyzed in the dictionary constructing phase. Thirdly, the proposed BSW model is performed at Arduino to detect faults mentioned in the first and second phases.

Otherwise, wireless sensor networks (WSNs) will transmit a fault code via one byte in hexadecimal coded form.

V. Experimental Setup and Parameter Setting

A. Experimental setup

1) Perform experiments: The experimental modules are shown in Fig. 2. To obtain faulty conditions, the following experiments are performed:

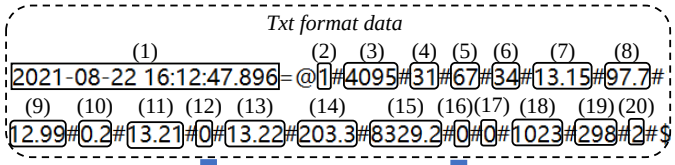
- Normal condition of the SIL-IoTs node (F0);
- Disconnect the power line of selected modules to obtain OC faults (F1, F3, F5, F6, F7, F11, and F12);
- Shorting the power line of rain fall detector to obtain SC fault (F9);
- Installing damaged modules to obtain lamp damage fault (F2);
- Shading solar panel or light intensity sensor to obtain mismatch between both of them (F8);
- Contacting the rain fall detector with water or conductive objects under sunny days to obtain rain fall detector faults (F10);
- Completely enclosed electrical box (F13);
- Wrong wiring of wind speed and direction sensor (F14 and F15);
- Equipping the clock chip with a low remaining energy coin cell battery to achieve clock synchronization after restarting the clock chip (F16).

The SC faults of battery, lure lamp, and solar panel are not performed for the following reasons: 1) these experiments can be hazardous to experimenters, and 2) SC faults occur quickly and are not responsive when detected. The description of experimental data is shown in Fig. 11, where flag bits are used to check the integrity of collected data. All data are acquired in decimal format except for timestamps. "=@", "#", and "$" are adopted to separate different measurements. Partial data have been uploaded online [50] and are shown in Fig. 12.

2) Comparison method: In our experiments, two fault self-detection methods designed for outdoor IoT modules are compared as follows.

- **TinyD2 [26]:** detecting faulty sensors via cumulative sum at a sliding window. Then, the bootstrap strategy is adopted to derive the confidence level (90%). If an apparent change is decided, the faulty condition is detected by localizing the change point of the cumulative sum at a sliding window.
- **TrusDet [25]:** recognizing faulty sensors via a fused result from a sliding window, in which a closer previous result has a greater impact on the data fusion.

Both methods obtain a final result from neighbor nodes' voting result (sink-level). Since the unreliable data transmission and high transmission delay in SIL-IoTs scenarios, we only consider both methods' sensor-level fault detection phase. The sliding window of both methods is set as 8 (same as the proposed method). For TinyD2, six reordered sliding windows by bootstrap are adopted as mentioned in [26]. For TrusDet, the fusion coefficient α is set as 0.125 according to the length of the sliding window.



Item	Description	Item	Description
(1)	Time	(11)	Lamp voltage
(2)	Node ID	(12)	Lamp current
(3)	Flag bits	(13)	Solar panel voltage
(4)	DHT11 temperature	(14)	Solar panel current
(5)	DHT11 related humidity	(15)	Light intensity
(6)	DS18B20 temperature	(16)	Voltage pulse
(7)	Power supply voltage	(17)	Audio frequency
(8)	Power supply current	(18)	Rain control
(9)	Metal mesh voltage	(19)	Wind direction
(10)	Metal mesh current	(20)	Wind speed

Fig. 11. Data description of experimental data, where 17 measurements are represented from (4) to (20).

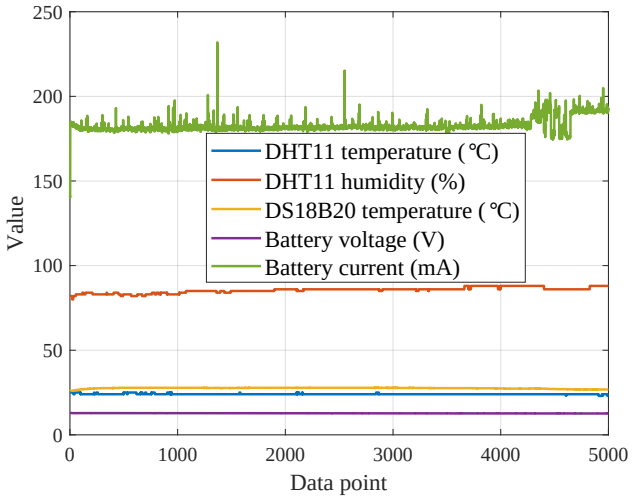


Fig. 12. An example of partial experimental data, where five measurements of all measurements are illustrated by 5000 data points.

In addition, several typical machine learning methods are compared as follows.

- **KNN:** K-nearest neighbor.
- **LR:** logistic regression.
- **SGD:** Stochastic gradient descent.
- **NB:** Naive Bayes.
- **SVM:** Support vector machine.
- **GBDT:** Gradient boosting decision tree.
- **RF:** Random forest.

The above machine learning methods are compiled by Python 3.8, and the parameters are set as default values in scikit-learn library [51].

3) Evaluation indicator: We employ several indicators to evaluate the performance of the different methods. For a classification issue, TP, FP, TN, and FN are widely used to describe the classification result of the proposed model:

- **TP:** True positive denotes that the prediction label is positive and the real value is positive.
- **FP:** False positive denotes that the prediction label is positive, but the real label is negative.
- **TN:** True negative denotes that the prediction label is negative and the real value is negative.
- **FN:** False negative denotes that the prediction label is negative, but the real label is positive.

Based on TP, FP, TN, and FN, the following evaluation indicators are adopted:

1) **Accuracy (Acc):** indicates the proportion of correct results predicted by the model, as expressed in 4.
2) **False alarm rate (FAR):** indicates the proportion of negative samples divided into positive samples in all negative samples, as expressed in 5. There is no negative sample divided into positive samples in all negative samples when FAR equals 0.
3) **Missing alarm rate (MAR):** indicates the proportion of positive samples divided into negative samples in all positive samples, as expressed in 6. There is no positive sample divided into negative samples in all positive samples when MAR equals 0.
4) **Receiver operating characteristic (ROC) curve:** is widely used to identify the performance of methods visually, which is plotted with the true positive rate (TPR, expressed in 7) as the ordinate and false positive rate (FPR, expressed in 8) as the abscissa [20]. The ROC curve near the upper left corner represents the most accurate results. In addition, the area under curve (AUC) is the area enclosed by the coordinate axis under the ROC curve, where the classifier with a larger AUC denotes better performance.
5) Moreover, model size and running time of methods are applied to estimate the computational costs and detection delay of different methods.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

$$FAR = \frac{FP}{TN + FP} \tag{5}$$

$$MAR = \frac{FN}{TP + FN} \tag{6}$$

$$TPR = \frac{TP}{TP + FN} \tag{7}$$

$$FPR = \frac{FP}{FP + TN} \tag{8}$$

B. Parameter setting

1) Matter-element of SIL-IoTs node: Matter-element vectors of SIL-IoTs' modules are proposed based on the above experiments, which is the basis of fault detection for SIL-IoTs nodes. It is the first work setting faulty features' parameters of SIL-IoTs nodes. All matter-element vectors of SIL-IoTs are shown in Table V, where subscripts of R are the items of Fig. 11. For instance, $R_{(4,5)}$ denotes the matter-element vector of DHT11. The subscripts of each characteristic represent the fault labels of Table IV, e.g., H_{F0} denotes the fault-free state of DHT11 humidity. $T^{(0,1)}$ denotes the difference of T^0 and T^1.

2) Correlation analysis: For detecting the mismatch between multi-variables, Pearson correlation coefficient [52] (denoted by r) is used to judge the correlation between different measured values. For a given sample size n for two variables X_i and X_j, the r can be expressed as:

$$r = \frac{\sum_{k=1}^{n}(x_{i,k} - \overline{X_i}) * (x_{j,k} - \overline{X_j})}{\sigma_i * \sigma_j} \tag{9}$$

Where σ_i and σ_j are the standard deviation of of X_i and X_j. $\overline{X_i}$ and $\overline{X_j}$ denote the average value of X_i and X_j. The correlation coefficients of 17 kinds of measurement are shown in Fig. 13, where high correlation ($|r| \geq 0.8$) and middle correlation ($0.5 \leq |r| \leq 0.8$) are highlighted. All measurements are at a statistical significance level (P0.01). The result indicates high correlations between voltages and currents of four modules.

3) Linear regression analysis for light intensity and solar panel's current: From Fig. 13, there is a high correlation value (0.99) between light intensity and current of the solar panel. Thus, we adopt the linear regression analysis method to build a mathematical model with only four floating-point units. The result of linear regression is shown in Fig. 14, where R^2 under the parameters $a = 0.0316$, $b = -6.28$ is 99.7% (close to 1 denotes high degree of regression). 3200 mA is the full load condition of the solar panel. Therefore, the current value will not increase in proportion to the light intensity value when the current value is over 3200 mA. The correlation between light intensity and the current of the solar panel can be expressed as follows:

$$y = \begin{cases} 0.0316x - 6.28, & x < 100000 \\ 3200, & x \geqslant 100000 \end{cases} \tag{10}$$

The threshold is set as 450 mA according to the measurements and correlation. Thus, the discriminant conditions can be expressed as:

$$f = \begin{cases} 0, & |\, y_{est} - y_{obs} \,| \geqslant 450 \\ 1, & |\, y_{est} - y_{obs} \,| < 450 \end{cases} \tag{11}$$

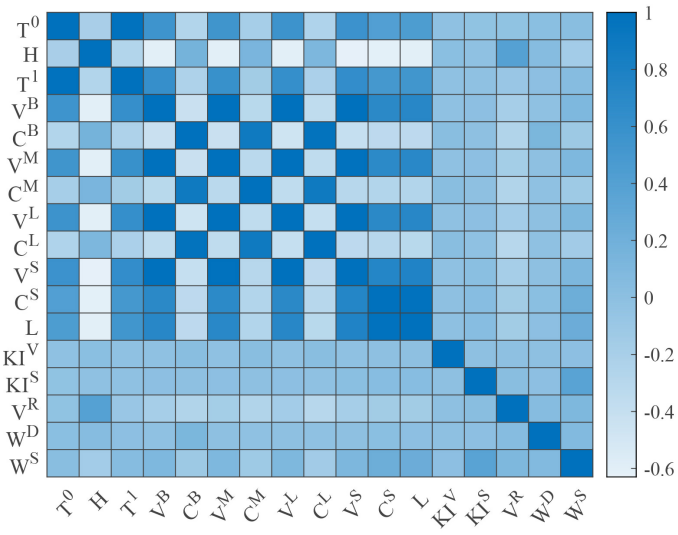


Fig. 13. Correlation between different measured values, where a value close to 1 indicates a high positive correlation, and a value close to -1 indicates a high negative correlation. The results indicate a high positive correlation between V^B,V^M,V^L, and V^S. Besides, the high correlation of C^S and L contributes to detecting F8.

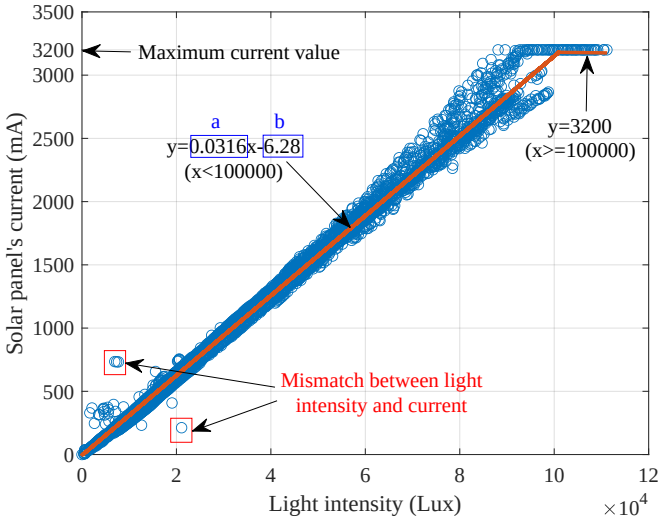


Fig. 14. The correlation between light intensity (independent variable x) and solar panel's current value (dependent variable y). The rated power of the adopted solar panel is 60 W. The maximum current value of a solar panel is 3200 mA. Therefore, the current value will not increase with the increase of the light intensity value if the rated power of the solar panel is reached.

In particular, there occurs an F7 fault when the light intensity value equals 0, but the solar panel's current value is greater than 0. Four floating point units are 0.0316, 6.28, 3200.0, and 450.0.

4) Correlation between rain fall detector and light intensity: According to the description of F10, we select light intensity values for a rainy day, shown in Fig. 15. The threshold is set according to the maximum value (20653.8 lux). In this way, the F10 fault will be detected if the light intensity value exceeds the maximum value on rainy days.

TABLE V
MATTER-ELEMENT VECTORS OF SIL-IOTS' MODULES

$$R_{(4,5)} = \begin{bmatrix} DHT11, \mathrm{T}^{0}_{F0}, (0,50) \\ \mathrm{H}_{F0}, (5,95) \\ \mathrm{H}_{F11}, 0 \\ \mathrm{T}^{0}_{F11}, 0 \end{bmatrix} \quad R_{(6)} = \begin{bmatrix} DS18B20, \mathrm{T}^{1}_{F0}, (0,60) \\ \mathrm{T}^{1}_{F12}, 85 \\ \mathrm{T}^{(0,1)}_{F13}, (9,) \end{bmatrix} \quad R_{(7,8)} = \begin{bmatrix} Battery, \mathrm{V}^{B}_{F0}, (10,14.5) \\ \mathrm{C}^{B}_{F0}, (55,200) \\ \mathrm{V}^{B}_{F6}, [14.5,) \\ \mathrm{V}^{M}_{F6}, [14.5,) \\ \mathrm{V}^{L}_{F6}, [14.5,) \end{bmatrix}$$

$$R_{(9,10)} = \begin{bmatrix} Metal\ mesh, \mathrm{V}^{M}_{F0}, (10,14.5) \\ \mathrm{C}^{M}_{F0}, (60,200) \\ \mathrm{C}^{M}_{F3}, [0,2] \\ \mathrm{KI}^{S,V}_{F4}, (20,) \end{bmatrix} \quad R_{(11,12)} = \begin{bmatrix} Lamp, \mathrm{V}^{L}_{F0}, (10,14.5) \\ \mathrm{C}^{L}_{F0}, (600,900) \\ \mathrm{C}^{L}_{F1}, [0,2] \\ \mathrm{C}^{L}_{F2}, (50,100) \end{bmatrix} \quad R_{(13,14)} = \begin{bmatrix} Solar\ panel, \mathrm{V}^{S}_{F0}, (10,14.5) \\ \mathrm{C}^{S}_{F0}, (0.1,3200) \\ \mathrm{V}^{S}_{F5}, [0,2] \end{bmatrix}$$

$$R_{(15)} = \begin{bmatrix} Light\ intensity, \mathrm{L}_{F0}, (0,150000) \\ \mathrm{L}_{F7}, 0 \end{bmatrix} \quad R_{(16)} = \begin{bmatrix} Voltage\ count, \mathrm{KI}^{V}, [0,255) \end{bmatrix} \quad R_{(17)} = \begin{bmatrix} Sound\ count, \mathrm{KI}^{S}, [0,255) \end{bmatrix}$$

$$R_{(18)} = \begin{bmatrix} Rain\ fall, \mathrm{V}^{R\_rain}_{F0}, (500,900) \\ \mathrm{V}^{R\_no}_{F0}, [900,1023] \\ \mathrm{V}^{R}_{F9}, 0 \end{bmatrix} \quad R_{(19)} = \begin{bmatrix} Wind\ direction, \mathrm{W}^{D}_{F0}, [0,30] \\ \mathrm{W}^{D}{}_{F14}, (30,) \end{bmatrix} \quad R_{(20)} = \begin{bmatrix} Wind\ speed, \mathrm{W}^{S}_{F0}, [0,5] \\ \mathrm{W}^{S}_{F15}, (5,) \end{bmatrix}$$

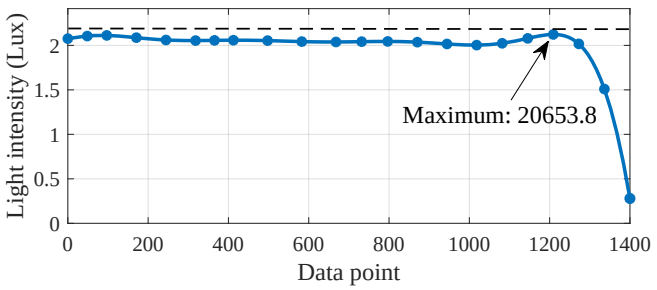


Fig. 15. An example of light intensity values on a rainy day. The maximum light intensity is set as 20653.8 lux according to the long history data, indicating that the light intensity value should not exceed 20653.8 lux on rainy days.

VI. PERFORMANCE DISCUSSION

A. Detection performance of different number of zero

The BSW method detects fault according to the number of zeros in the sliding window, which indicates that increasing the number of zeros can reduce the false alarm rate. To verify this hypothesis, we compared the fault detection strategies using one to five zeros, named BSW-1, BSW-2, BSW-3, BSW-4, and BSW-5. The number followed by 'BSW-' represents the number of zeros required for detecting a fault.

The ROC curve of 10-fold cross-validation results are shown in Fig. 16. Due to the multi-classification of faults, macro-averaging is applied to calculate each fault label's average TPR and FPR. Obviously, when the number of zeros increases from 1 to 5, the BSW method performs better, showing multiple zeros' optimization effects. The main reason for this phenomenon is that as the number of zeros increases, the BSW method will detect faults more cautiously, thus reducing the misclassification probability of positive and negative samples under each category. The ROC curve results of the BSW-4 and BSW-5 are slightly better than that of the BSW-3 because more zeros may lead to unnecessary caution in the BSW method.

Average FAR, and MAR results under 10-fold cross-validation for each fault are illustrated in Fig. 17 (a) and (b). As the number of zeros increases, fewer fault-free samples (positive) are classified as faulty samples (negative); thus, the FAR decreases significantly. Meanwhile, the MAR increases because more faulty samples are divided into fault-free samples. Therefore, according to the scenario's requirements, the appropriate number of zeros can be selected based on the FAR and MAR. As shown in Fig. 17 (a), for some specific faults (e.g., F3, F5, F8, F12), the performance of the BSW-1 and BSW-2 on FAR is obviously insufficient to meet the accuracy requirements. In addition, the MAR values of the BSW-1 and BSW-2 are unstable especially for F0 and F16 in Fig. 17 (b). Therefore, better fault detection results can be achieved by using 3-5 zeros rather than 0 or 1 zero.

The trade-off between detection accuracy, reliability, and delay is critical for performing the fault detection scheme on IoT devices [53]. As illustrated in Fig. 17 (c) and (d), average detection accuracy and detection delay increase with the increasing number of zeros. In particular, the BSW-1 and BSW-2 show a significant decrease in accuracy when detecting F0, suggesting that using 3-5 zeros is a better choice. However, compared with the BSW-3, the BSW-4 and BSW-5 have no significant advantages on the FAR, MAR, ACC, and ROC curve. With a less diagnostic delay, according to Fig. 17 (d), three zeros are applied and will be the default setting in the following section. The minimum detecting latency is obtained by the cooperation of multiple features (explained in Fig. 8). The maximum detecting latency is obtained by only a single feature estimation.

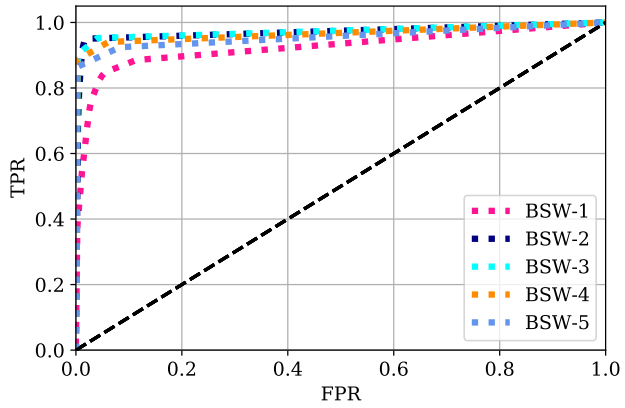


Fig. 16. The ROC curve of the different zero numbers used for fault detection, where the number followed "BSW-" denotes the zero used. Notably, BSW-3, BSW-4, and BSW-5 have the same performance.

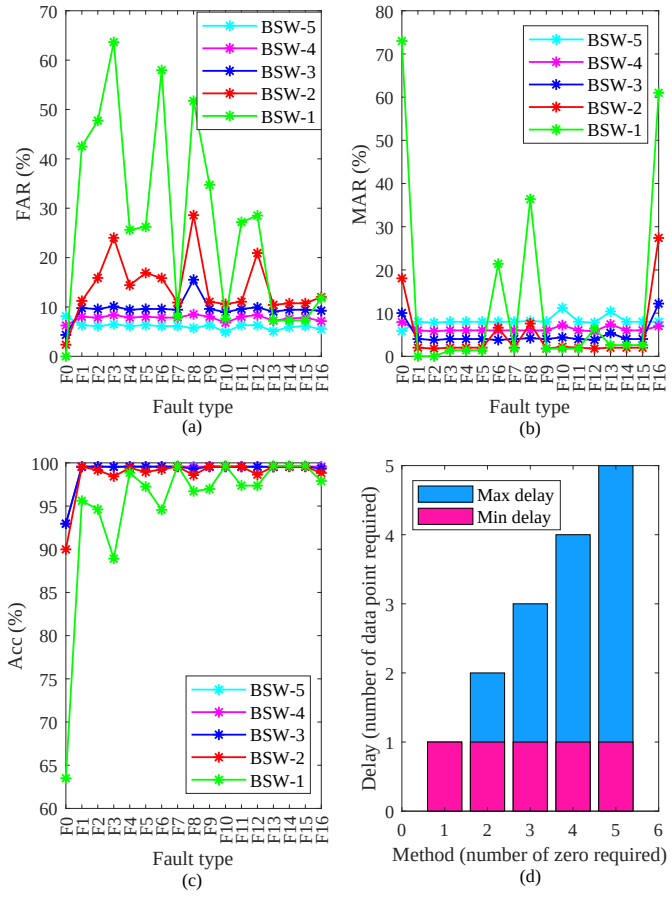


Fig. 17. Performance of the different numbers of zeros used for detection. (a), (b) and (c) denote the false alarm rate, missing alarm rate, and accuracy for different fault types under the number of zeros from 1 to 5, respectively. The red and blue bars in (d) denote the minimum and maximum delay under the number of zeros from 1 to 5.

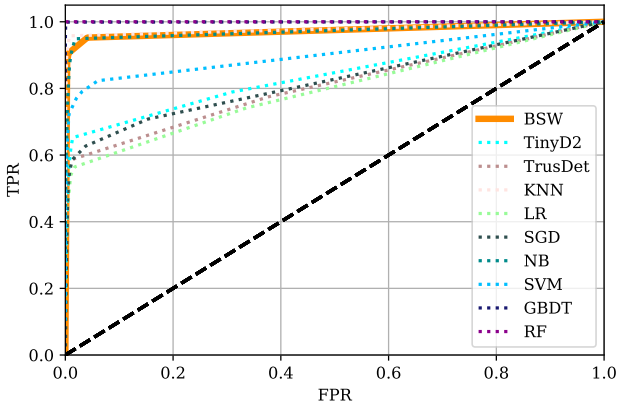


Fig. 18. The ROC curve of different methods for fault detection, where the AUC of KNN, GBDT, and RF methods are superior to that of the BSW method.

B. Detection performance of different methods

In this section, the detecting performance of the BSW method, two prior knowledge-based methods (TinyD2 and TrusDet), and seven typical machine learning methods are validated under all types of faults.

The ROC curves of all methods are shown in Fig. 18, where the ROC curve of the BSW method is depicted by a solid line. Obviously, the BSW method's performance is better than most methods. Specifically, the TinyD2, TrusDet, LR, and SGD methods have poor ROC curves. TinyD2 and TrusDet methods obtain poor performance because they detect faults only by one feature rather than the correlation of elements. Thus, they cannot achieve high TPR for mismatch of two features (i.e., F4, F8, F10, F13, and F16). Moreover, both LR and SGD methods are vulnerable to noise data and are biased towards the category with the superior number in the training samples if the sample is unbalanced. Therefore, LR and SGD methods fail to detect faults because fault data represents only a small fraction of the total data in practice. Notably, GBDT and RF achieve the highest performance because they have significant generalization performance.

The average detection acc, FAR, and MAR under 10-fold cross-validation are shown in Fig. 19 (a), (b), and (c), respectively. Due to the same reason, the acc, FAR, and MAR of the TinyD2 and TrueDet methods are inferior to that of the BSW method. In particular, compared with the performance of the TinyD2 and TrusDet, the BSW method obtains a lower average FAR of 2.56% and 1.82%, which demonstrates that the multiple zero mechanism of the BSW method is superior to that of the other two methods in reducing false alarms. Moreover, It is shown that the KNN, GDBT, and RF methods have better performance compared with other methods, where the RF method even has 100% acc, 0% FAR, and 0% MAR. Therefore, the RF method is best when considering fault detection accuracy. However, considering the complexity of the models for Arduino, although the performance of the BSW method is slightly inferior to that of the RF method, the performance of the BSW method is acceptable.

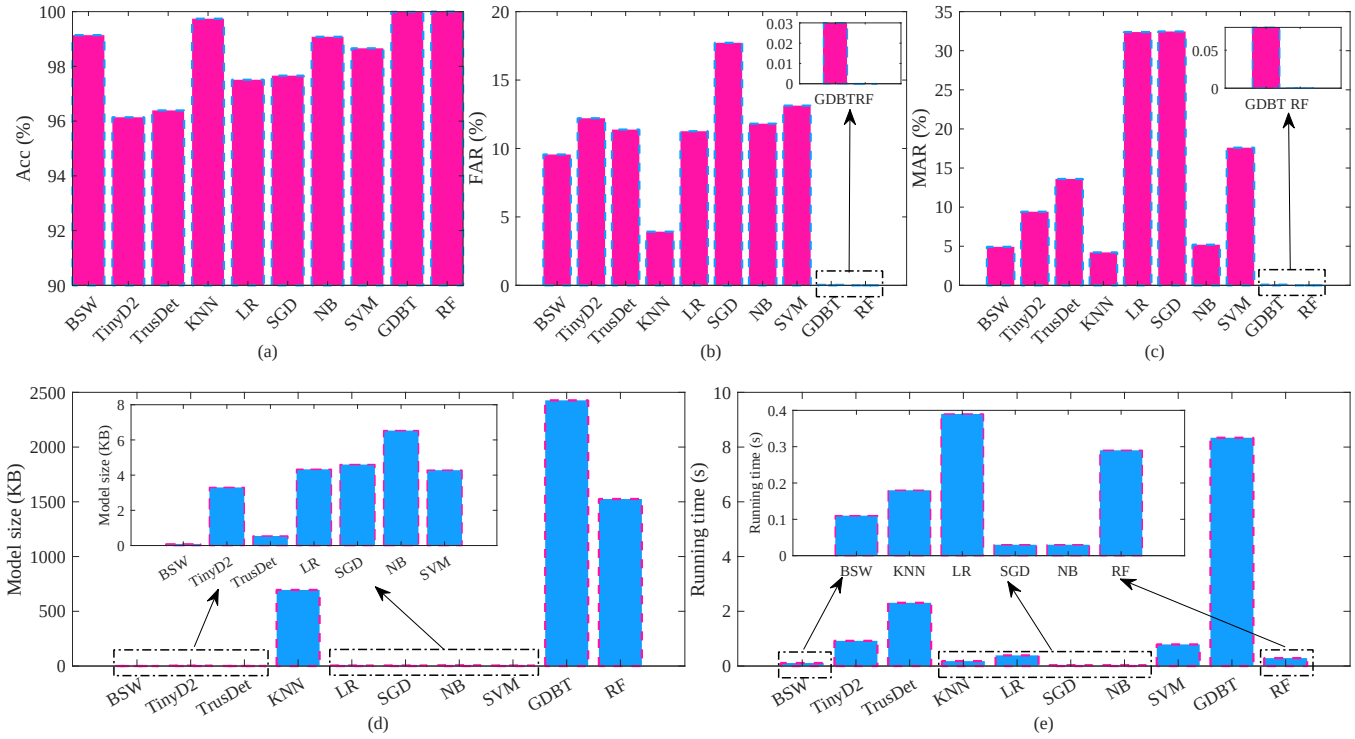


Fig. 19. The average performance of different methods. (a), (b), (c), (d), and (e) denote the detection accuracy, false alarm rate, missing alarm rate, model size, and running time of different methods. The results indicate that the BSW method has high detection accuracy and low false alarm rate, missing alarm rate, model size, and running time.

C. Computational cost of different methods

This section explores the computational cost of different methods. The results were conducted using Python 3.8 on a computer equipped with an Intel Core i5-11260H processor (2.6GHz) and 16GB RAM. The model size (number of parameters) and average running time are demonstrated in Fig. 19 (d) and (e). The model sizes of the BSW, TinyD2 and TrusDet methods are calculated by the required parameters and temporary variables. For instance, there are 14 integers and five floating point parameters (4B for each) and 16 temporary variables (denoting sliding window for different faults, 1B for each one). Therefore, the model size of the BSW method can be expressed as (14+5)*4+16*1=92B. The model sizes of machine learning methods are calculated by the file size of the saved trained model.

The results indicate that the model size of the BSW method is significantly smaller than the other methods. Besides, the average running time of the BSW method is 0.11s, which is slightly lower than most other methods except for SGD and NB methods. Because SGD and NB methods have lower detection accuracy, the BSW method is more suitable for fault self-detection of SIL-IoTs. In addition, although the GBDT and RF methods have better fault detection accuracy than the BSW method, they both require a much larger system cache and longer running time than the BSW method. In summary, the BSW method saves a tenfold system cache and has an acceptable fault detection delay without significantly decreasing detection accuracy.

D. Energy consumption

In this section, we compiled the BSW method in C++ language and applied it to the Arduino of PCB to evaluate the energy consumption of the BSW method at a real-world SIL-IoTs node. We repeated the experiment 10 times, and each experiment lasted 100 seconds. The voltage and current values of the PCB are obtained by a voltage and current sensor.

The energy consumption of PCB with no detection module and self-detection module are shown in Fig. 20. The result indicates that the BSW method only increases the energy consumption by less than 1% according to the average energy consumption, which rises from 320.222 mW to 321.468 mW. Because the BSW method only performs simple threshold judgments based on the acquired data and then performs multiple "SRCAND" operations, it does not consume too many system resources of the Arduino, thus avoiding a significant increase in energy consumption.

Besides, a hexadecimal fault code listed in Table IV and node ID (5B) will be transmitted to the cloud rather than all float and int data (76B as the format in Fig. 11). Thus, 71B data are no longer required to be transmitted, which saves a lot of data forwarding and transmission energy consumption. Note that when multiple faults occur, only one fault code is issued. Thus, multiple faults do not increase the amount of data sent. In summary, based on the detection performance results, computational cost, and energy consumption, it is acceptable to add the BSW method to SIL-IoTs' Ardunio for fault self-detection.

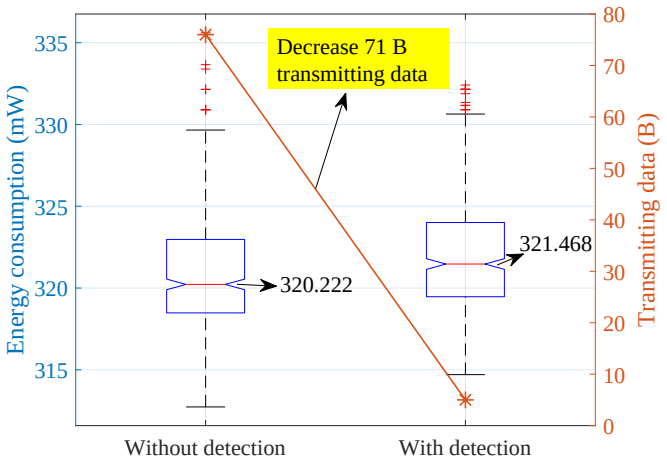


Fig. 20. The box plot represents energy consumption with and without the fault self-detection module. The brown line represents the required transmitting data. The results indicate that adopting the fault self-detection module only increases about 0.39% energy consumption but reduces about 93.42% transmitting data.

E. Advantage and Limitation

1) Advantage:

- **Improving fault detection performance:** Fig. 16 and Fig. 17 illustrate that the number of zeros impacts the performance of the BSW method, which denotes that fault detection sensitivity of the BSW method can be adaptive according to the requirements of the scenario. In addition, from Fig. 19, the proposed method can detect faults with an average accuracy of 99.14%, an average false alarm rate of 9.56%, and an average missing alarm rate of 4.92%. Considering the low frequency of failures and that the fault self-detection scheme is implemented on resource-constrained devices, the BSW method can meet the accuracy requirement of SIL-IoTs fault detection compared with the other two prior knowledge-based methods.
- **Saving system cache in resource-constrained IoT devices:** The BSW method stores eight consecutive states as binary values into one byte, greatly reducing the cost of performing time series analysis. Therefore, although the accuracy of partial machine learning methods is superior to that of the BSW method, the BSW method costs the smallest system cache (92B) and a lower fault detection delay (0.11s). Since most SIL-IoTs nodes are equipped with only one Arduino chip as the processor, such a small resource consumption reduces the pressure to develop other software functions.
- **Enhancing the practicability of the fault self-detection model:** For a real-world SIL-IoTs node, the BSW method only costs an additional average of 0.39% energy consumption. It reduces 71B transmitting data by reducing 93.42% energy consumption for each data forwarding. They also show that in the field of WSNs, the BSW method can be applied as an abnormal detection scheme with significance in reducing data transmission and energy consumption.

2) Limitation: Although the proposed method has advantages in the trade-off between accuracy, false positive rate, model size, and running time, it still has some limitations.

- Due to the microcontroller's limited data storage and computing capacities, historical data will not be stored. Although the mismatch fault can be detected, the faulty module cannot be detected in this case. Therefore, the root cause of the mismatch fault has to rely on the information of neighbor nodes. Selecting appropriate neighbor nodes is a challenge that may be solved by spatial correlation analysis. In addition, optimizing the energy consumption and data transmission of SIL-IoTs nodes while ensuring detection accuracy is a key issue.
- The detection threshold for each fault is preset when using a prior knowledge for constructing a fault self-detection model. Therefore, the original rules may no longer be applicable if there is a great environmental change. One solution is to set the detection threshold according to adaptive methods.

VII. Conclusion

In this paper, we presented the new hardware design of the SIL-IoTs node to reduce the electromagnetic inference conditions, avoid an unexpected shutdown case, and improve the fault detection mechanism. Moreover, to perform fault detection on the resource-constrained SIL-IoTs node, we proposed the BSW fault self-detection method to detect faults timely and reduce data transmission. The BSW method could detect a fault on resource-constrained devices only by a few bytes, where each byte was applied as eight bits for storing binary time-series information to reduce the false alarm rate. To evaluate the performance of the BSW method, we performed several experiments and compared them with two prior knowledge-based methods (outdoor scenario) and seven baseline machine learning algorithms. Conclusions are summarized as follows.

1) Three zeros had demonstrated its effectiveness for reducing the detection delay while it keeps the high accuracy, low false alarm rate, and low missing alarm rate. It provided a solution for selecting detecting sensitivity according to specific scenarios.
2) The proposed method was applied to SIL-IoTs and showed advantages in the balance of accuracy, false alarm rate, detecting delay, and usage of memory.
3) The framework of the proposed method could also be applied to other scenarios with similar characteristics to perform the abnormal detection scheme on resource-constrained devices. In particular, the BSW method does not rely on neighbor node information, so it is suitable for scenarios with untrustworthy neighbor nodes.

In the future, we plan to investigate the faulty conditions in that the node itself cannot detect faults will be studied via distributed fault diagnosis approaches. Future works will include: a) analyzing the spatial correlation between neighbor nodes to obtain local evidence, b) providing an adaptive detecting threshold for adapting to changes in the environment, c) double-checking the fault after fault self-detection scheme to ensure the faulty conditions.

REFERENCES

[1] F. Yang, L. Shu, K. Huang, K. Li, G. Han, and Y. Liu, "A partition-based node deployment strategy in solar insecticidal lamps Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 11 223–11 237, 2020.

[2] K. Huang, K. Li, L. Shu, X. Yang, T. Gordon, and X. Wang, "High voltage discharge exhibits severe effect on ZigBee-based device in solar insecticidal lamps Internet of Things," *IEEE Wireless Communications*, vol. 27, no. 6, pp. 140–145, 2020.

[3] M.-S. Liao, C.-L. Chuang, T.-S. Lin, C.-P. Chen, X.-Y. Zheng, P.-T. Chen, K.-C. Liao, and J.-A. Jiang, "Development of an autonomous early warning system for bactrocera dorsalis (hendel) outbreaks in remote fruit orchards," *Computers and Electronics in Agriculture*, vol. 88, pp. 1–12, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S016816991200172X

[4] M. W. de Carvalho, E. R. Hickel, B. Bertoldi, G. C. Knabben, and Y. R. d. Novaes, "Design of a smart led lamp to monitor insect populations in an integrated pest management approach," *Revista Brasileira de Engenharia Agrícola e Ambiental*, vol. 25, pp. 270–276, 2021.

[5] K. Li, L. Shu, K. Huang, Y. Sun, F. Yang, Y. Zhang, Z. Huo, Y. Wang, X. Wang, Q. Lu, and Y. Zhang, "Research and prospect of solar insecticidal lamps internet of things," *Smart Agriculture*, vol. 1, no. 3, pp. 13–28, 2019, (in Chinese with English abstract).

[6] F. Yang, L. Shu, Y. Yang, Y. Liu, and T. Gordon, "Improved coverage and connectivity via weighted node deployment in solar insecticidal lamp internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 10 170–10 186, 2021.

[7] F. Yang, L. Shu, Y. Yang, G. Han, S. Pearson, and K. Li, "Optimal deployment of solar insecticidal lamps over constrained locations in mixed-crop farmlands," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 13 095–13 114, 2021.

[8] X. Guo, L. Shu, X. Yang, E. Nurellari, K. Li, B. Du, and H. Yao, "Two-hop energy consumption balanced routing algorithm for solar insecticidal lamp internet of things," *Sensors*, vol. 22, no. 1, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/1/154

[9] S. Shao, Q. Zhang, S. Guo, L. Sun, X. Qiu, and L. Meng, "Intelligent farm meets edge computing: Energy-efficient solar insecticidal lamp management," *IEEE Systems Journal*, pp. 1–10, 2022.

[10] X. Yang, L. Shu, K. Huang, K. Li, Z. Huo, Y. Wang, X. Wang, Q. Lu, and Y. Zhang, "Characteristics analysis and challenges for fault diagnosis in solar insecticidal lamps Internet of Things," *Smart Agriculture*, vol. 2, no. 2, pp. 17–27, 2020, (in Chinese with English abstract).

[11] O. López, M. M. Rach, H. Migallon, M. P. Malumbres, A. Bonastre, and J. J. Serrano, "Monitoring pest insect traps by means of low-power image sensor technologies," *Sensors*, vol. 12, no. 11, pp. 15 801–15 819, 2012.

[12] X. Yang, L. Shu, J. Chen, M. A. Ferrag, J. Wu, E. Nurellari, and K. Huang, "A survey on smart agriculture: Development modes, technologies, and security and privacy challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 273–302, 2021.

[13] X. Yang, L. Shu, K. Huang, K. Li, and H. Yao, "Poster abstract: Insecticidal performance simulation of solar insecticidal lamps Internet of Things using the number of falling edge trigger," in *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2021, pp. 1–2.

[14] H. Kai, S. Lei, L. Kailiang, Y. Xing, Z. Yan, W. Xiaochan, and S. Qin, "Design and prospect for anti-theft and anti-destruction of nodes in solar insecticidal lamps internet of things," *Smart Agriculture*, vol. 3, no. 1, p. 129, 2021.

[15] H. Zheng, R. Wang, Y. Yang, Y. Li, and M. Xu, "Intelligent fault identification based on multisource domain generalization towards actual diagnosis scenario," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 2, pp. 1293–1304, 2020.

[16] A. H. Sabry, F. H. Nordin, A. H. Sabry, and M. Z. Abidin Ab Kadir, "Fault detection and diagnosis of industrial robot based on power consumption modeling," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 9, pp. 7929–7940, 2020.

[17] H. Yan, Y. Xu, F. Cai, H. Zhang, W. Zhao, and C. Gerada, "PWM-VSI fault diagnosis for a PMSM drive based on the fuzzy logic approach," *IEEE Transactions on Power Electronics*, vol. 34, no. 1, pp. 759–768, 2019.

[18] M. Zhu, J. Li, W. Wang, and D. Chen, "Self-detection and self-diagnosis methods for sensors in intelligent integrated sensing system," *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 247–19 254, 2021.

[19] A. Moon, X. Zhuo, J. Zhang, and S. W. Son, "Ad¡sup¿2¡/sup¿: Improving quality of iot data through compressive anomaly detection," in *2019 IEEE International Conference on Big Data (Big Data)*, 2019, pp. 1662–1668.

[20] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, 2021.

[21] H. Darvishi, D. Ciuonzo, and P. S. Rossi, "Real-time sensor fault detection, isolation and accommodation for industrial digital twins," in *2021 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, vol. 1, 2021, pp. 1–6.

[22] D. Haldimann, M. Guerriero, Y. Maret, N. Bonavita, G. Ciarlo, and M. Sabbadin, "A scalable algorithm for identifying multiple-sensor faults using disentangled rnns," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 3, pp. 1093–1106, 2022.

[23] S. Lu, R. Ma, T. Sirojan, B. Phung, and D. Zhang, "Lightweight transfer nets and adversarial data augmentation for photovoltaic series arc fault detection with limited fault data," *International Journal of Electrical Power Energy Systems*, vol. 130, p. 107035, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014206152100274X

[24] T. Yu, X. Wang, and A. Shami, "Recursive principal component analysis-based data outlier detection and sensor data aggregation in iot systems," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2207–2216, 2017.

[25] S. He, J. Chen, Y. Shu, X. Cui, K. Shi, C. Wei, and Z. Shi, "Efficient fault-tolerant information barrier coverage in internet of things," *IEEE Transactions on Wireless Communications*, vol. 20, no. 12, pp. 7963–7976, 2021.

[26] K. Liu, Q. Ma, W. Gong, X. Miao, and Y. Liu, "Self-diagnosis for detecting system failures in large-scale wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5535–5545, 2014.

[27] K.-H. Chao and C.-T. Chen, "A remote supervision fault diagnosis meter for photovoltaic power generation systems," *Measurement*, vol. 104, pp. 93–104, 2017.

[28] Z. Chen, Y. Cao, S. X. Ding, K. Zhang, T. Koenings, T. Peng, C. Yang, and W. Gui, "A distributed canonical correlation analysis-based fault detection method for plant-wide process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2710–2720, 2019.

[29] Z. Liu, Z. Huang, J. Wang, C. Yue, and S. Yoon, "A novel fault diagnosis and self-calibration method for air-handling units using Bayesian inference and virtual sensing," *Energy and Buildings*, vol. 250, p. 111293, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378778821005776

[30] B. Gou, Y. Xu, Y. Xia, G. Wilson, and S. Liu, "An intelligent time-adaptive data-driven method for sensor fault diagnosis in induction motor drive system," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9817–9827, 2019.

[31] M. Schmid, E. Gebauer, C. Hanzl, and C. Endisch, "Active model-based fault diagnosis in reconfigurable battery systems," *IEEE Transactions on Power Electronics*, vol. 36, no. 3, pp. 2584–2597, 2021.

[32] Z. Zhao, Y. Chen, Y. Yu, M. Han, C. Tang, and C. Yao, "Equivalent broadband electrical circuit of synchronous machine winding for frequency response analysis based on gray box model," *IEEE Transactions on Energy Conversion*, vol. 36, no. 4, pp. 3512–3521, 2021.

[33] B. Li, Q. Hu, Y. Yu, and G. Ma, "Observer-based fault-tolerant attitude control for rigid spacecraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 5, pp. 2572–2582, 2017.

[34] A. Li, G.-R. Duan, M. Liu, and J. b. Fu, "Fault-tolerant quantized sliding mode observers design for a class of takagi-sugeno fuzzy system with unmeasurable premise variable," *IEEE Transactions on Fuzzy Systems*, pp. 1–1, 2021.

[35] M. Van, S. S. Ge, and D. Ceglarek, "Fault estimation and accommodation for virtual sensor bias fault in image-based visual servoing using particle filter," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 4, pp. 1312–1322, 2018.

[36] R. Xiong, Q. Yu, W. Shen, C. Lin, and F. Sun, "A sensor fault diagnosis method for a lithium-ion battery pack in electric vehicles," *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 9709–9718, 2019.

[37] Y. Yu, Y. Zhao, B. Wang, X. Huang, and D. Xu, "Current sensor fault diagnosis and tolerant control for VSI-based induction motor drives," *IEEE Transactions on Power Electronics*, vol. 33, no. 5, pp. 4238–4248, 2018.

[38] C. Sun, X. Chen, R. Yan, and R. X. Gao, "Composite-graph-based sparse subspace clustering for machine fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 1850–1859, 2020.

[39] Y. Tang, C. Wang, M. Wang, H. Hao, and J. Zhao, "Based on self-learning dictionary circuit board fault diagnosis device," in *2017 IEEE*

2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2017, pp. 2653–2657.
[40] A. Xenophontos, J. Rarey, A. Trombetta, and A. M. Bazzi, "A flexible low-cost photovoltaic solar panel emulation platform," in *2014 Power and Energy Conference at Illinois (PECI)*, 2014, pp. 1–6.
[41] K. Huang, H.-G. Stratigopoulos, S. Mir, C. Hora, Y. Xing, and B. Kruseman, "Diagnosis of local spot defects in analog circuits," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 10, pp. 2701–2712, 2012.
[42] U.-M. Choi, J.-S. Lee, F. Blaabjerg, and K.-B. Lee, "Open-circuit fault diagnosis and fault-tolerant control for a grid-connected NPC inverter," *IEEE Transactions on Power Electronics*, vol. 31, no. 10, pp. 7234–7247, 2016.
[43] Y. Chen, W. Li, F. Iannuzzo, H. Luo, X. He, and F. Blaabjerg, "Investigation and classification of short-circuit failure modes based on three-dimensional safe operating area for high-power IGBT modules," *IEEE Transactions on Power Electronics*, vol. 33, no. 2, pp. 1075–1086, 2018.
[44] Q. An, J. Li, X. Ma, G. Bi, and S. Bian, "An effective method to evaluate the battery life under unreliable mains associated with rigorous high temperature," in *2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC)*, 2011, pp. 1–3.
[45] "IEC 60896-21:2004," [Online]. Available: https://webstore.iec.ch/publication/3850, accessed: Dec. 14, 2021.
[46] J. D. Bošković and R. K. Mehra, "A decentralized fault-tolerant control system for accommodation of failures in higher-order flight control actuators," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 5, pp. 1103–1115, 2010.
[47] C. Yang, J. Yang, Z. Liu, and S. Tian, "Complex field fault modeling-based optimal frequency selection in linear analog circuit fault diagnosis," *IEEE Transactions on Instrumentation and Measurement*, vol. 63, no. 4, pp. 813–825, 2014.
[48] Y. Li, Z. Teng, C. Liang, and J. Li, "Detection and localization of short-duration variations using sliding window SVD and sparse signal decomposition," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 9, pp. 6912–6920, 2020.
[49] A. S. Edun, C. LaFlamme, S. R. Kingston, H. V. Tetali, E. J. Benoit, M. Scarpulla, C. M. Furse, and J. B. Harley, "Finding faults in PV systems: Supervised and unsupervised dictionary learning with SSTDR," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4855–4865, 2021.
[50] "Partial data of the SIL-IoTs fault self-detection scheme," [Online]. Available: https://github.com/harryyangx/Fault-self-detection-for-SIL-IoTs, Feb. 2022, Accessed: Apr. 13, 2022.
[51] X. Xu, X. Yan, C. Sheng, C. Yuan, D. Xu, and J. Yang, "A belief rule-based expert system for fault diagnosis of marine diesel engines," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 2, pp. 656–672, 2020.
[52] W. Xu, C. Chang, Y. S. Hung, and P. C. W. Fung, "Asymptotic properties of order statistics correlation coefficient in the normal cases," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2239–2248, 2008.
[53] Z. Zhang, A. Mehmood, L. Shu, Z. Huo, Y. Zhang, and M. Mukherjee, "A survey on fault diagnosis in wireless sensor networks," *IEEE Access*, vol. 6, pp. 11 349–11 364, 2018.

Xing Yang (Student Member, IEEE) received the M.S. degree in control engineering from Nanjing University of Information Science and Technology, China, in 2018. He is currently working toward the Ph.D. degree in agricultural electrification and automation from College of Engineering, Nanjing Agricultural University, Nanjing, China.

He has been a Visiting Ph.D. Student with the University of Lincoln, Lincoln, U.K., since November 2021. He has authored or co-authored papers in several prestigious journals/conferences, such as IEEE Wireless Communications, IEEE/CAA Journal of Automatica Sinica, and IEEE Transactions on Industrial Informatics. His current research interests include fault diagnosis in wireless sensor networks, agricultural Internet of Things reliability, and machine learning algorithm.

Lei Shu (Senior Member, IEEE) received the B.S. degree in computer science from South Central University for Nationalities, Wuhan, China, in 2002, the M.S. degree in computer engineering from Kyung Hee University, Seoul, South Korea, in 2005, and the Ph.D. degree in computer engineering from the Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland, in 2010.

Until 2012, he was a Specially Assigned Researcher with the Department of Multimedia Engineering, Graduate School of Information Science and Technology, Osaka University, Osaka, Japan. He is currently a Distinguished Professor with Nanjing Agricultural University, Nanjing, China, and a Lincoln Professor with the University of Lincoln, Lincoln, U.K. He is also the Director of the NAU-Lincoln Joint Research Center of Intelligent Engineering, Nanjing, China.

Kailiang Li received the B.S. degree from Guangdong University of Petrochemical Technology, Maoming, China, in 2015. He is currently a Research Assistant with the College of Engineering, Nanjing Agricultural University, Nanjing, China. He has published over 10 papers in related conferences, and journals in the areas of sensor networks and Internet of Things. His research interests are wireless sensor networks and Internet of Things.

Zhiqiang Huo received the B.Sc. degree and M.Sc. degree from the School of Information Engineering, University of Geosciences (Beijing), China in 2013 and 2016, respectively. He received his Ph.D. degree from the University of Lincoln, UK, in 2020. He now is a Research Associate with the Institutional Research Information System (IRIS) at the University College London (UCL), UK. He is also a member of CHIMERA at UCL, which dedicates to improving the healthcare and outcomes of critically ill patients. His research interests include time-series complexity analysis, data mining, machine learning, and artificial intelligence. He has published over 30 publications in peer-reviewed IEEE leading journals and conferences, such as IEEE Transactions on Instrumentation and Measurement, IEEE Network, and IEEE ACCESS.

Sheng Shu is currently pursuing the B.S.degree in electronic information science and technology from College of Artificial Intelligence, Nanjing Agricultural University, Nanjing, China. His research interests include electronic circuit design, microcontroller program, and signal processing.

Edmond Nurellari (Member, IEEE) has been a Faculty Member with the School of Engineering, University of Lincoln, Lincoln, U.K., since April 2017, where he is currently an Associate Professor of Electrical Engineering and Robotics. His research interests include machine learning, robotics for communications, distributed signal processing, signal processing on graphs, resource allocations, and distributed decisions in WSNs. He is an Invited Reviewer of the IEEE Transactions on Signal and Information Processing over Networks, the IEEE Communication Letter, the Springers Wireless Networks Journal, the Springers Digital Signal Processing Journal, and the IEEE Flagship conferences. Over the past few years, he is the Guest Editor of the Special Issue Smart Agricultural Applications with Internet of Things for Sensors Journal, an Editorial Board of the Frontiers in Communications and Networks Journal, the Journal Topics Board for Journal of Sensor and Actuator Networks, TPC Member of the IEEE iSES, International Conference on Smart and Sustainable Agriculture ((SSA'2021)), and a Reviewer for several UKRI grants, including EPSRC and Future Leaders Fellowship. He was the recipient of the Carter Prize for the best Ph. D. thesis, titled Distributed Detection and Estimation in Wireless Sensor Networks: Resource Allocations, Fusion Rules, and Network Security, in the School in the year 2017–18, University of Leeds, Leeds, U.K.