IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 3, 2021, accepted September 21, 2021, date of publication September 27, 2021, date of current version October 8, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115819

A Spatiotemporal and Multivariate Attribute Correlation Extraction Scheme for Detecting Abnormal Nodes in WSNs

NESRINE BERJAB, HIEU HANH LE, AND HARUO YOKOTA, (Senior Member, IEEE)
Tokyo Institute of Technology, Tokyo 152-8550, Japan

Corresponding author: Nesrine Berjab (berjab@de.cs.titech.ac.jp)

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP 16007.

ABSTRACT Many heterogeneous sensors exhibit strong spatio-temporal correlations that can be used to enhance the abnormal node detection problem in a wireless sensor network (WSN). Corruption in these correlations has been shown effective in detecting false data injection attacks. In this paper, we adopt a new cross-correlation-based method to extract the sensor relationships. It utilizes the observed spatiotemporal (ST) and multivariate-attribute (MVA) sensor correlations to decide whether the sensor is subject to abnormalities or represents actual events. Based on the ST correlations, the cross-correlation is extracted in both space and time by conducting shape-based logical subclustering and two-phase analysis methods. In the first analysis phase, the system uses a variable-size sliding window and a median absolute deviation (MAD) measure. If the collected sensor data streams output a certain percentage of anomalous points, the MAD will flag these points as anomalous measurements, and all the sensor data and the window size will be passed to the second analysis phase. The latter performs both tumbling-window and sliding-window analyses to extract multicriteria cross-correlation measures. Finally, all the extracted sensor time-series features will be fed to the shape-based clustering to generate a sensor similarity-like graph. The latter reflects the similarity degree of the sensor with the other nodes. The nodes with a low similarity degree below the threshold will be identified as candidate abnormal nodes. Based on the observed MVA correlations, a set consisting of a few rules is introduced to check whether the detected candidate abnormal nodes represent actual events. Finally, if abnormal nodes exist, then such nodes are reported. Our experiments using two real-world datasets demonstrate that our proposed approach detects abnormal nodes with an average accuracy of 96.50%, an average precision of 88.69%, and a recall rate of 93.00%.

INDEX TERMS Anomaly detection, Internet of Things, sensor data correlations, wireless sensor networks, false data injection attacks.

I. INTRODUCTION

Since the emergence of the Fourth Industrial Revolution, there has a growing trend to use elements of the Internet of Things (IoT), such as mobile phones, Bluetooth low-energy beacons, and wireless sensor networks (WSNs). The IoT can be described as a dynamic and distributed networked system that uses wireless connectivity and comprises a wide range of uniquely identifiable embedded computer-like devices. Such devices' primary requirements are to monitor their environmental conditions, report sensor data, and perform appropriate actions in response to the surrounding circumstances [1].

A. MOTIVATION

However, the accuracy of such decisions depends upon the reliability and trustworthiness of the collected sensor data. Unfortunately, the environment of sensor networks makes these issues more challenging. Indeed, the primary defect with IoT security lies in the fact that the sensors are somewhat basic, and their operational software or firmware is usually poorly coded [1], [2]. The accelerated deployment of IoT technology has often resulted in postponing security issues and relegating them to secondary importance, and

The associate editor coordinating the review of this manuscript and approving it for publication was Oussama Habachi.

some device manufacturers have concentrated on profitability at the expense of security. It has been stated [3] that 85.00% of IoT developers are urged to bring a product to market before adequate security can be implemented. Moreover, some studies [4]–[10] have shown that attackers can manipulate sensor data in real deployments and cause catastrophic damage because of the lack of security consciousness in these systems. Because of their exposure to hostile environments and their inherent limitations, sensors are often subject to failure and exhibit various vulnerabilities. It is then easy for attackers to compromise some sensor nodes' reliability and manipulate the integrity of the sensed data through false data injection attacks (FDIAs), for instance. Such failures and compromised sensors hamper the system's functionality, leading to inappropriate decisions by operators and possibly leading to catastrophic effects.

To guarantee a safe and reliable IoT system, the security around its constituent devices should be examined more closely. This is where anomaly detection is becoming a necessity. Detecting anomalies in sensor data streams is an important area of research and has been a subject of much interest. It aims to uncover abnormal yet interesting knowledge that does not conform to normal behavior and may therefore indicate suspicious behavior. Various sensor anomaly detection algorithms have been proposed [11]–[14], but few are able to address the problem of FDIAs. Most anomaly detection methods proposed for the IoT, particularly for WSNs, focus only on specific types of network attacks.

One traditional way to detect these kinds of attacks is to deploy an estimator and a detector in the system controller [15]. The job of the estimator consists of estimating and calculating the possible future readings and comparing them with the actual readings. The detector then triggers an alarm whenever there is a significant difference between the estimated and actual readings.

However, FDIAs are not directly detectable by such a traditional approach [15]–[17]. The attacker can recognize the usual conditions of the monitored environment and can easily inject false data into the regular sensor readings without being detected by the system, thereby misleading operators into making inappropriate decisions. Moreover, the current primitive data detection and analysis techniques were designed to deal only with faults and failures rather than those associated with malicious activity. In addition, most current techniques for anomaly detection only consider the content of the data source itself without considering the correlation context of the data. Heterogeneous sensor data tend to exhibit a strong correlation in both space and time that can be used to enhance the abnormal node detection problem in WSNs. Corruption in these correlations is very likely to be affected by the presence of anomalies.

B. CONTRIBUTIONS

The present paper is an extended version of our prior work [18], in which the spatiotemporal (ST) and multivariate-attribute (MVA) correlations of heterogeneous sensor readings are considered in the detection process. The collected sensor data were analyzed by computing the cross-correlation between heterogeneous sensor streams. The cross-correlation helps align two time series when one is lagged with respect to others. The computed lag correlation is then compared against two predefined thresholds: the lag threshold and the correlation threshold.

However, there are two minor issues in the method presented in the original paper. These issues lie in the assumed naive threat scenario and the ST correlation extraction method. There is a possibility that the attacker may inject measurements different from the observed ones, but this will not be easily detectable because the data describe wrong measurements that are still within the customary circumstances. Under such conditions, experiments have shown that our prior work could not detect the attack and correctly characterize the compromised sensors. Moreover, experimental results have shown that the detection accuracy is sensitive to the choice of threshold parameters. The cross-correlation function (CCF) estimation accuracy on the sensor data streams' correlations increases when the analyzed time series is long.

To tackle these limitations, we aim to extract the ST correlations between the sensor nodes that can effectively detect abnormal nodes while reducing the false alarm rates. To achieve this aim, in this paper, we extend our prior work by adding additional processing to the original ST correlation extraction process. We also introduce a new threat model that generates various attack patterns, which allows us to test the detection algorithm and evaluate its performance against different threat severity levels. We also added detailed descriptions of the proposed detection method, evaluations, and further discussion, which complement our prior work. The main contributions of this paper are as follows.

1) We introduce a new ST-based correlation extraction method to efficiently detect the abnormal sensor nodes generated from the newly considered threat model.
2) We propose a new attack strategy to generate a malicious dataset from the original sensor data, which allowed us to test the detection algorithm and evaluate its performance against different threat severity levels. We create evaluation data, including various FDIA patterns and missing data, based on the initially collected dataset.
3) We demonstrate the effectiveness of our proposed method by conducting a variety of performance evaluations. Compared with our prior work, we also augment our evaluation with an additional larger-scale dataset. Our experiments using the two real-world datasets demonstrate that our proposed method detects abnormal nodes with an average accuracy of 96.50%, an average precision of 88.69%, and a recall rate of 93.00%.

C. ORGANIZATION OF THE PAPER

The rest of the paper is organized as follows. In Section II, we review the prior methods of anomaly detection in WSNs. Section III describes some essential background characteristics before introducing our proposed method. Section IV presents the detailed architecture and design of our proposed method to detect abnormal nodes. We then describe the experimental setup in Section V and present an analysis of the results and an evaluation in Section VI. Finally, Section VII contains some concluding remarks and perspectives.

II. RELATED WORKS

Sensor anomaly detection has received a considerable amount of attention in the literature. It refers to identifying instances or unusual events or observations that raise suspicions. Nevertheless, it is often difficult to discern the sensor nodes' anomalies from the actual anomalies that emerged from the monitored environment.

In this context, the type of deployed WSN, the adopted anomaly detection methodology, and the type of anomalies of interest may significantly impact the design methodology. In this paper, we categorize existing related works into three orthogonal research directions related to anomaly detection in WSNs.

- **Sensor node types**: Homogeneous WSNs (i.e., one-type sensors) vs. heterogeneous WSNs (i.e., multitype sensors).
- **Detection methods**: Methods directly running on sensing devices (i.e., distributed methods) vs. methods running on the cloud (i.e., centralized methods)
- **Anomaly types**: Anomalies spanning a short period of time (i.e., short-term anomalies) vs. anomalies spanning a long period of time (i.e., long-term anomalies)

A. SENSOR NODE TYPES

Anomaly detection in homogeneous WSNs has received much attention in the literature. Most of these methodologies [19]–[25] deploy multiple one-type sensors to detect abnormal nodes. In this case, the homogeneous device is analyzed based on the fact that neighboring same-type sensors are often correlated and tend to generate similar measurements. For instance, [19] proposed a method of neighborhood data fusion in decentralized anomaly detection. In [20], the authors proposed a trust evaluation model that can detect the state of a node according to the data trust. In [21], the authors combined both statistical and machine learning techniques to detect anomalies in the network behavior of IoT devices. The solution is based on constructing behavioral device templates. In [22], the authors compared the sensor measurement against the predicted measurements by using the time-series forecasting method to detect faulty sensors.

All of the presented works only consider homogeneous-based anomaly detection approaches. Furthermore, these approaches are usually based on complex mathematical analysis and statistical methods applied to sensor data and tailored to the specific numerical characteristics of the considered type of sensor. Thus, applying such methods to heterogeneous sensors may not be straightforward.

All of the presented works only consider a homogeneous-based approach to resolving sensor failure problems or detecting intrusions. Moreover, the cost of deployment is high because of the redundancy of using the same type of sensors spatially close to each other.

More recently, a new wave of research driven by heterogeneous WSN paradigms has emerged and gained rapid uptake. The concept behind the heterogeneous-based detection approach is when the sensor network combines different types of sensor data to detect anomalies. Nevertheless, this concept is still regarded as not mature, and some challenging issues need to be addressed before deploying it in actual WSN environments. An approach for monitoring heterogeneous WSNs and identifying hidden correlations between heterogeneous sensors was proposed by [23]. This approach can identify the hidden correlation between heterogeneous sensors but has not been specifically conceived for anomaly detection.

The proposed approach in [24] detects and identifies faulty devices proposed for smart homes. The authors used a context-based method to detect faulty heterogeneous nodes. Their experiments showed that their proposed approach successfully detects and identifies faulty devices in a short detection time. However, they do not consider the fact that sensor nodes can be subject to attacks and cause abnormal nodes in the network. In [25], SMART was proposed as a sensor failure detection system based on classifier outputs. The classifier is trained to recognize the normal activity patterns based on different subsets of sensors. In [26], 6thSense is a proposed context-aware intrusion detection system for heterogeneous sensors. It monitors the changes in sensor data by creating a contextual model to distinguish normal and malicious sensor nodes.

While these heterogeneous-based approaches efficiently detect anomalies, they require additional development. Most of these data detection and analysis techniques were designed to only deal with faults and failures and not coordinated malicious activity. Moreover, many rely on background knowledge and labeled training data. In this paper, we develop a framework to detect anomalies in heterogeneous WSNs.

B. DETECTION METHODS

Anomaly detection in WSNs can be classified as either methods directly running on sensing devices (i.e., distributed methods) or methods running on the cloud (i.e., centralized methods). Performing anomaly detection in a central processing system allows us to adopt complex algorithms and, consequently, to obtain accurate results. In [18], a centralized-based approach is proposed where all heterogeneous sensor streams are collected and controlled in a centralized base station. The proposed solution evaluates the intensity of the correlation between the sensor streams by calculating the lag correlation between them. In [26], the authors propose a centralized failure detection approach where the base station aggregates the network sensor readings and detected failures by finding an insufficient flow of incoming data.

TABLE 1. Comparison between approaches.

	Sensor node types		Detection methods		Anomaly types	
	Homogeneous	Heterogeneous	Distributed	Centralized	Short	Long
[19, 20]	yes	no	yes	no	no	yes
[21, 22]	yes	no	no	yes	yes	no
[24-26]	no	yes	no	yes	no	yes
[27, 28]	no	yes	yes	no	yes	no
Prior method [18], [23]	no	yes	no	yes	no	yes
Proposed method	no	yes	no	yes	yes	yes

In contrast, distributed methods run directly on sensor nodes equipped with light computation capability. Most of these approaches require historical data samples to be kept in the sensor node, which has limited memory storage. In [27], [28], a rule-based distributed fuzzy inference system for WSNs was proposed that combines both local and neighboring observations to identify the occurrence of events. In this paper, our framework is centralized-based to overcome the drawbacks of the distributed method and guarantee good detection accuracy.

C. ANOMALY TYPES

Another essential aspect to consider regarding sensor anomaly detection is the time span covered by the anomaly itself. Most of the works mentioned [18]–[28] effectively detect short-term anomalies or long-term anomalies, but not both. For instance, in [18], a time-lagged cross-correlation analysis was used to extract these relationships. The correlation between the sensor data streams is captured by computing the CCF between them. However, the CCF estimation accuracy on the sensor data streams' correlations increases when the analyzed time series is long. Indeed, the method is more effective for detecting anomalies spanning a long period of time. Anomalies spanning a short period of time cannot be detected, as their occurrence does not affect long-term scale correlation. It is well known that the length of the analyzed sensor data streams may influence the correlation between the sensors and lead to false alarms. In other words, the method in [18] depends considerably on the sensor streams' length and the period of anomaly records, which often varies between the sensors.

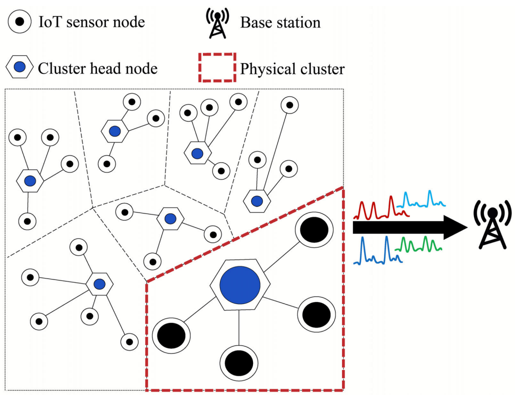


FIGURE 1. A typical WSN architecture.

TABLE 2. Comparison between approaches in terms of requirements.

	R1	**R2**	**R3**
[19, 20]	no	no	yes
[21-22]	no	no	no
[24, 25]	yes	no	no
Prior method [18], [23, 26-28]	yes	no	yes
Proposed method	yes	yes	yes

Table 1 recapitulates the characteristics of each mentioned related work, including the proposed method in this paper. To summarize the works, while they can efficiently detect anomalies, they require further development.

First, most of these data detection and analysis techniques were designed to deal only with short-term anomalies or long-term anomalies. Second, many techniques rely on user intervention for labeling training data or supplying additional background information, which is a time-consuming and challenging task in itself. A solution claiming to be adequate should satisfy the following requirements.

- **R1) Practicability** is the requirement for the system to consider the use of various types of heterogeneous sensors in the deployed network.
- **R2) Multitarget anomaly detection** is the requirement for the system to consider the detection of both short-term and long-term anomalies to guarantee a reliable IoT system.
- **R3) Feasibility** is the requirement whereby the monitoring system does not require training data to undertake the anomaly detection process.

Table 2 summarizes the extent to which each of the previously mentioned works meet these requirements, together with our proposed method in this paper.

With these identified requirements, in this paper, we propose a new cross-correlation-based method to decide whether the sensor is subject to anomalies (both short-term and long-term anomalies) or represents actual events. In particular, the ST and MVA correlations of heterogeneous sensor readings are considered in the detection process. Our proposed method satisfies all the requirements listed in Table 2.

III. PRELIMINARY BACKGROUND

This section provides the essential background characteristics used in our proposed framework and discusses some assumptions about the monitoring environments considered in this paper.

A. SYSTEM AND SENSOR DATA MODEL

An environmental monitoring application in a WSN is defined as an application that monitors the real world and issues a report whenever an event of interest occurs during a certain period in a specific location. The typical WSN architecture we consider (Figure 1) consists of heterogeneous sensor nodes, a base station (BS), and a network connecting all sensor nodes. The BS is a server for collecting and processing sensor data. All the sensor nodes in the WSN are connected to this BS directly or indirectly.

This paper addresses the network scalability issue by adopting a cluster-based network topology. Indeed, we adopt the clustering method as a network architecture not only because it saves the sensor nodes' energy but also because it allows the capturing of the correlation between the sensors, enhances the trustworthiness of the system, and improves the detection rate. Certainly, a cluster-based routing algorithm in WSNs is another approach that allows sensor data aggregation to reduce the overall communication cost. However, in this paper, we do not aggregate the collected data. The sensor data are collected from each cluster in the WSN and forwarded to the BS. Indeed, the objective of this paper is to detect abnormal nodes in WSNs. Thus, we need to collect every sensor stream in the centralized node and analyze whether there is corruption in their correlation.

To implement a thorough monitoring system, n sensor nodes $(S_1, S_2, \ldots, S_n)$ are geographically grouped into physical clusters, each covering a certain area. In each physical cluster, one elected node is chosen to be a cluster head node (CH). Other nodes in the cluster are called sensor nodes (SNs), and they report their sensed readings to their CH. Once all the sensed data within the cluster are collected, the CH forwards the messages directly to the BS.

Depending on the application, a CH node can be a special sensor with more potential than other sensor nodes in terms of energy, bandwidth, and memory. However, most of the proposed systems for WSNs based on environmental monitoring in the literature assume that all the sensor nodes in WSNs are inexpensive and equal in terms of computation, communication, and power. In addition, it has been stated that the existence of sensors with different capacities raises many technical issues, especially in terms of data routing. Therefore, to guarantee a domain-independent application, in this paper, we consider that all the sensor nodes in the network have the same performance characteristics. In other words, even though the CH acts as a particular node within the physical cluster, it still acts as an SN. This implies that apart from its role in forwarding readings from SNs to the BS, its readings are also included in the computations for its own cluster. Besides, the role of the CH is periodically rotated among all nodes in order to balance the energy consumption and the traffic load in the network. We assume that each sensor sends its reading to the CH, and then CH resends the reading to the BS. As a result, each reading is sent twice, except for the CH's readings. Furthermore, each cluster should include heterogeneous sensors for sensing and collecting data about a variety of attributes, such as temperature, humidity, and light intensity.

Each sensor node (i.e., SN and CH) S_i has a unique identifier, where $i \in [1, n]$. Each S_i is characterized by five attributes, L, T, Cp, Cl, and an output stream O.

- Let $L(S_i)$ be the location of sensor S_i, specified by its geographic coordinates x_i, y_i, and z_i.
- Let D be the set of sensor types, where D includes *Temperature*, *Humidity*, *Light*, *Smoke*, etc.
- Let $T(S_i)$ be the node's sensor type, where $T(S_i) \in D$.
- As mentioned, the clustering concept is adopted for the network topology. Although several complex and innovative clustering techniques have been proposed for WSNs, this paper considers a very simple clustering technique for environmental monitoring in WSNs. In addition, the role of the CH is periodically rotated among all nodes to balance the energy consumption and the traffic load in the network. We denote $size(Cp)$ as the number of sensors deployed in the physical cluster Cp. Let $Cp(S_i)$ be the physical cluster within which S_i is located. The clustering formation is based on a defined distance threshold, th_d. Two sensors, S_i and S_j, belong to the same cluster Cp if and only if $Cp(S_i) = Cp(S_j)$ and the distance between $L(S_i)$ and $L(S_j)$ is less than th_d.
- This paper adopts a subclustering procedure based on the sensor's spatial correlation to separate the physical clustering further and guarantee accurate correlation extraction. SNs within a physical cluster having similar structural patterns are then grouped into a subcluster identified as a logical cluster. We denote $size(Cl)$ as the number of sensors deployed in the logical subcluster Cl. Let $Cl(S_i)$ be the logical subcluster of $Cp(S_i)$ within which S_i is located. Two sensors, S_i and S_j, belong to the same logical subcluster Cl if and only if $Cl(S_i) = Cl(S_j)$, $Cp(S_i) = Cp(S_j)$, and S_i and S_j have strong spatial data correlations.
- Finally, let $O(\vec{S_i})$ be sensor S_i's data stream, where $O(\vec{S_i}) = \{O(S_i, 1), \ldots, O(S_i, t), \ldots, O(S_i, m)\}$. $O(S_i, t)$ is the node's output data stream with every S_i sensing data at time t, and m is the length of the sensor data stream.

B. SENSOR DATA ANOMALIES

Wireless SNs have limited resources and are often exposed to a hostile environment. Therefore, the collected sensor data may be distorted by anomalies, which can be classified into two categories. Some anomalies correspond to fail-stop anomalies, where the sensor stops generating values after failure (e.g., the device completely shuts down). The second category corresponds to nonfail-stop faults, which occur unpredictably and appear when a device exhibits abnormal behavior and generates and reports incorrect values. In this case, we describe the sensor readings as a combination of real value and error terms. The error terms can be either a random error or systematic error. Random errors are errors that fluctuate around the real value because of noise or because of limitations of the sensor in terms of precision. The second type is systematic error, which has a nonzero mean. It usually shifts the value away from the real value. Such errors emerge because of different causes, such as faults or nodes that are compromised by external attackers. Moreover, in a real-world sensor network, the sensor data streams are nonstationary time series that may frequently change over time due to events in the observed environment. Thus, the generated sensor data may also differ significantly from its standard value.

A sensor reading is considered to be ideal when there are no systematic errors and can be described by the following equation:

$$O(S_i, t)' = O(S_i, t) + \epsilon, \tag{1}$$

where $O(S_i, t)'$ is the collected sensor measurement, $O(S_i, t)$ is the real value and ϵ is the random error, with $\epsilon \sim N(0, \sigma^2)$. Figure 2 displays the different anomalies that frequently occur in real-world sensor data.

- **Anomaly 1**: This is a short-term high anomalous value.
- **Anomaly 2**: This is a sequence of values that are unaffected by the input and remain the same.
- **Anomaly 3**: This is a continuous sequence of anomaly one.
- **Anomaly 4**: This represents missing values because of the absence of generated or reported sensed data.
- **Anomaly 5**: This is a short-term high value that goes beyond the expected degree of the normal measurement range.
- **Anomaly 6**: This is a sequence of multiple values of anomaly five that are greater than the expected range of normal measurements.

All these anomalies in the WSN data provide insight into what kind of anomalies may appear in the sensor data. However, the six defined anomalies are not comprehensive for all anomalies that may appear in sensor data. Furthermore, these types of anomalies can either span over a short or span over a long period of time in the collected sensor data streams.

Anomalies with different causes may have different characteristics. From the collected sensor data in Figure 2, we can observe that anomalies are abnormally generated values that spatially or temporarily differ from the standard values. Therefore, it is helpful to classify anomalies based on specific features.

Such features may contain how much the data deviate from standard data instances, the frequency of occurrences, and the time and location of anomalies within the WSN. The next section explains the rationale of identifying the cause of anomalies based on their correlation with other SNs.

C. SENSOR DATA CORRELATIONS

One approach to determining whether a particular sensor measurement is normal and identifying abnormal measurements is to explore the similarity relationships between the sensor measurements. These similarities are what we call sensor correlations that can be derived from the local sensor measurement with respect to other SNs in the network. In the presence of anomalous data, the correlations between the sensor measurements would be corrupted. ‘‘When’’, ‘‘where’’, and ‘‘what’’ are three key points for distinguishing abnormal SNs from normal SNs. By providing answers to these three questions, we can distinguish three types of sensor correlations: temporal, spatial, and multivariate-attribute correlations.

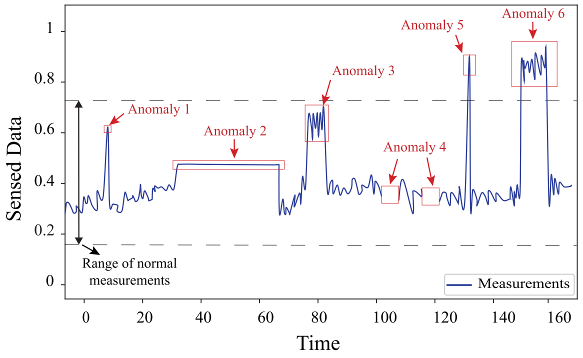


FIGURE 2. **Possible sensor data anomalies in real-world sensor data.**

1) WHEN: TEMPORAL CORRELATION

Temporal correlations occur between each consecutive observation of a sensor node. In other words, the sensed data tend to be the same as or similar to the readings observed at previous times. Moreover, the degree of change between consecutive sensor measurements is usually constrained by the temporal variation characteristics of the observed physical phenomenon. Sensed data that do not vary according to the observed environmental patterns imply that there are anomalous data.

2) WHERE: SPATIAL CORRELATION

Typical WSN applications require high-density sensor deployment to maintain good area coverage within the spatial domain. Thus, the measurements from multiple homogenous sensors located in the same field and monitoring the environment at the same time tend to show a high degree of similarity. As a result, neighboring sensor observations are highly correlated with the degree of correlation increasing with decreasing internode separation.

3) WHAT: MULTIVARIATE-ATTRIBUTE CORRELATION

Homogeneously sensed data usually contain both time and space information. However, the data generated by heterogeneous sensors are not independent, and we can obtain additional valuable information that may lead to better insights into the monitored environment. We may call this the ‘‘observed MVA.’’ In normal situations, when there is no interference from events or abnormal nodes, heterogeneous

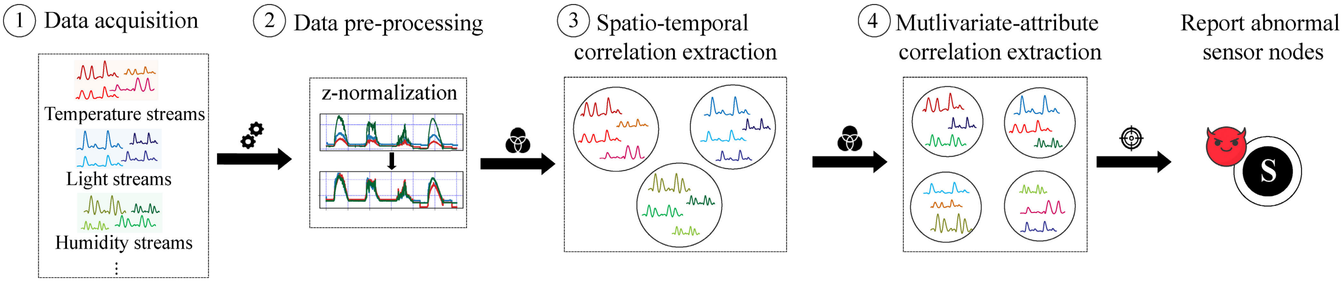

1 Data acquisition
Temperature streams
Light streams
Humidity streams
2 Data pre-processing
z-normalization
3 Spatio-temporal correlation extraction
4 Mutlivariate-attribute correlation extraction
Report abnormal sensor nodes
S

FIGURE 3. Workflow of the abnormal node detection framework overview.

sensor streams located in the same cluster tend to be correlated in both intracluster and intercluster senses, whereas the occurrence of events usually appears in specific observations. For example, a particular change in sensor data caused by an event will exhibit temporal and spatial correlation. This change will continue for a period of time after the event occurs. Moreover, if the event occurs in a specific cluster, then heterogeneous sensors located in the same cluster will show a high degree of correlation, unlike sensors located in other clusters. For instance, in a fire detection system, light intensity, temperature, and smoke density are all necessary elements of the information used to identify that a fire has been detected [18].

D. ASSUMPTIONS

Using correlations between heterogeneous sensor properties, our aim is to detect the abnormal nodes that generate anomalies to the collected sensor data streams. Our research is based on the following assumptions:

- To reduce the complexity of the problem, we assume that every sensing environment is characterized by its environmental conditions, such as temperature, light intensity, and relative humidity.
- All clusters must be composed of both homogeneous and heterogeneous SNs to maintain high event-detection accuracy.
- While some SNs may be compromised and considered abnormal nodes, we assume that the majority of the sensors will remain trustworthy.

IV. PROPOSED APPROACH

Our objective in this paper is to differentiate false alarms from valid alarms and guarantee the trustworthiness of the system by detecting abnormal nodes. This problem can be expressed as follows:

Problem: Given n coevolving correlated sensor-stream sequences provided by n heterogeneous sensors collected at the same time, determine, at any point in time, which SNs are abnormal, and report all such nodes.

We propose a novel framework to detect abnormal nodes in sensor network environments with heterogeneous sensor data to achieve our objective. The proposed framework is illustrated in Figure 3. To achieve our objective, we follow a four-step process. First, heterogeneous sensor data are collected from the various physical clusters. Before analyzing the sensor data stream cross-correlations between the sensors, there is a need for a preprocessing step. Afterward, the system extracts the ST correlation for each physical cluster by analyzing the cross-correlation between homogeneous sensor streams. Next, based on the background knowledge of the monitored environment and the observed MVA correlations, a number of rules are introduced to check whether abnormal nodes or real events have been detected. Finally, if abnormal nodes exist, then such nodes are reported.

A. STEP 1: DATA ACQUISITION

The first step involves collecting heterogeneous sensor streams from the various clusters deployed in the monitored area. After the data are collected, preprocessing can begin.

B. STEP 2: DATA PREPROCESSING

Before analyzing the sensor data stream cross-correlations between the sensors, there is a need for a preprocessing step. The measurements from homogenous sensors located in the same cluster and simultaneously monitoring the environment tend to show high observation similarity. Nevertheless, there are some exceptions that we should consider. Some sensor types are more sensitive to the observed environment than others. The interval scale of the measurements depends on the sensor location itself. For instance, the light intensity of the sensor located next to the window tends to be higher than that of the sensor located away from a light source. Such a difference in magnitude would corrupt the correlation information we would want to extract, as the introduced scale difference alters the true sensor cross-correlation to varying extents.

Therefore, there is a need to standardize the sensor data streams to make the magnitude and time scale uniform. To handle the different scales in amplitude, the sensor data streams are first standardized using *z-normalization*, also known as normalization to zero mean and unit of energy [29].

Let $\vec{O(S_i)} = \{O(S_i, 1), \ldots, O(S_i, t), \ldots, O(S_i, m)\}$ be an S_i data stream with m data points, and z-normalization of $\vec{O(S_i)}$ is defined as follows:

$$z\text{-}(\vec{O(S_i)}) = \frac{O(S_i, t) - \mu_{S_i}}{sd_{S_i}}, \tag{2}$$

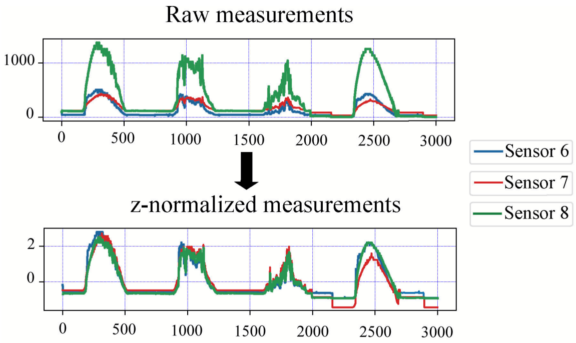


FIGURE 4. z-Normalization of sensor data streams deployed in our laboratory.

where

$$\mu_{S_i} = \frac{\sum_{t=1}^{T} O(S_i, t)}{m}, \tag{3}$$

$$sd_{S_i} = \sqrt{\frac{\sum_{t=1}^{m} \left|(O(S_i, t) - \mu_{S_i})^2\right|}{m}}, \tag{4}$$

where μ_{S_i} is the arithmetic mean of S_i's measurements from $O(S_i, 1)$ to $O(S_i, m)$ and sd_{S_i} is the standard deviation of all the S_i measurements in the given data stream $\overrightarrow{O(S_i)}$.

Normalization will cause the sensor data streams to be invariant to scale and offset. Figure 4 illustrates a real-world situation observed in a WSN deployed in our laboratory. Sensors six, seven, and eight are three light sensors located in the same physical cluster. However, because sensor eight is located near the window, the raw recorded light intensity is higher than that for the other two sensors. Nevertheless, sensor eight still maintains the same structural pattern as the other two sensors. After applying normalization to the raw light data streams, we can observe that their *z-normalized* versions are highly similar. Overall, this preprocessing is an essential step, as it will allow the anomaly detection method to focus on the structural similarities rather than on the amplitudes. To simplify notation in the rest of the paper, we shall write $(\overrightarrow{O(S_i)})$ in place of z-$(\overrightarrow{O(S_i)})$. This means that we do not refer to the raw measurements but the z-normalized measurements.

C. STEP 3: SPATIOTEMPORAL CORRELATION EXTRACTION

In our prior work [18], a novel approach was proposed to extract the ST and multivariate attribute correlations between heterogeneous sensors. The collected sensor data were analyzed by computing the cross-correlation between homogeneous sensor streams both within clusters (intracluster) and between clusters (intercluster). However, there are two minor issues in the method presented in the original paper. These issues lie in the assumed naive threat scenario and the ST correlation extraction method.

To tackle these limitations, we aim to extract the ST correlations between the SNs that can effectively detect abnormal nodes while reducing the false alarm rates. In this paper, we extend our prior work by adopting new methods to extract the ST correlation between the sensor streams to achieve this aim. The proposed detection methods learn how each sensor's data correlate within the sensor network, and abnormal nodes are identified by exploiting the anomalies in these correlations. Figure 5 illustrates the workflow of the ST correlation extraction. For each physical cluster, the system conducts shape-based logical subclustering. Each logical subcluster consists of homogeneous sensors sharing the same sensor patterns from the sensor data stream. Then, the system proceeds to the two-phase sensor data stream analysis. In the first analysis phase, the system uses a variable-size sliding window and an MAD measure to detect short-term anomalies. If the collected sensor data stream outputs a certain percentage of anomalous points, the MAD will flag these points as anomalous measurements, and all the sensor data and the window size will be passed to the second analysis phase. The latter performs both tumbling-window and sliding-window analyses to extract multicriteria cross-correlation measures to detect long- and short-term anomalies. Finally, all the extracted sensor time-series features will be fed to the shape-based clustering to generate a sensor similarity graph. The latter reflects the similarity degree of the sensor with the other nodes. The nodes with a low degree of similarity below the threshold will be reported as abnormal nodes. In the following sections, we will look at each step in more detail and the reason behind the choice of each adopted technique.

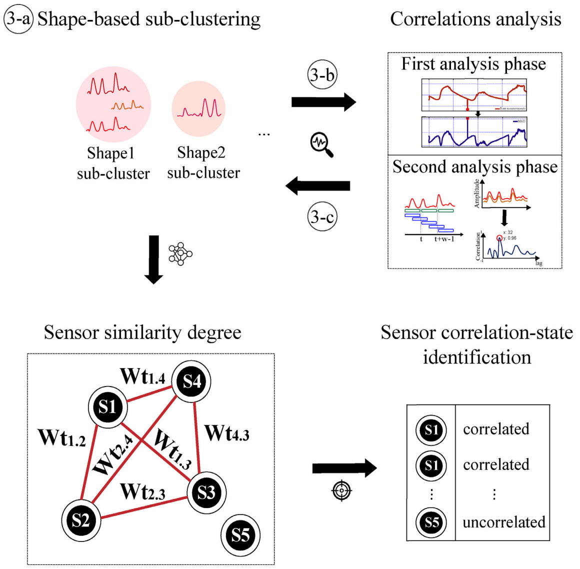


FIGURE 5. Spatiotemporal correlation extraction.

1) STEP 3-A: SHAPE-BASED SUBCLUSTERING

As mentioned in the previous subsection, it is assumed that neighboring sensors in the same physical cluster share similar dynamic environmental characteristics, but some may have events within their respective sensing range. Such a difference in timescale and magnitude would corrupt the correlation. Thus, the objective here is to divide the SNs within the physical cluster into logical subclusters having similar structural patterns. We use the k-shape [30] method for evaluating the sensor-stream data similarity to tackle this problem. This is a clustering method based on an iterative refinement similar to the one used in k-means. However, unlike k-means, it groups the time series into clusters based on their shape similarity, regardless of the amplitude and phase difference. Moreover, it is domain-independent, as it uses a distance based on coefficient-normalized cross-correlation, named shape-based distance (SBD)

Let $\overrightarrow{O(S_i)} = \{O(S_i, 1), \ldots, O(S_i, m)\}$ and $\overrightarrow{O(S_j)} = \left\{O(S_j, 1), \ldots, O(S_j, m)\right\}$ be two sensor data streams of length m of sensors S_i and S_j, respectively ($i \neq j$), and $T(S_i) = T(S_j)$ and $Cp(S_i) = Cp(S_j)$. To determine the similarity between $\overrightarrow{O(S_i)}$ and $\overrightarrow{O(S_j)}$, we use the SBD measure, which is defined as follows:

$$\begin{aligned} &SBD(\overrightarrow{O(S_i)}, \overrightarrow{O(S_j)}) \\ &= 1 - max_w \\ &\times \left(\frac{CC_w(\overrightarrow{O(S_i)}, \overrightarrow{O(s_j)})}{\sqrt{R_0(\overrightarrow{O(S_i)}, \overrightarrow{O(S_i)}) \cdot R_0(\overrightarrow{O(S_j)}, \overrightarrow{O(S_j)})}}\right), \end{aligned} \tag{5}$$

which takes a value between 0 and 2, with 0 indicating a high similarity between the sensor data streams. In every iteration, the method k-shape performs two steps in its centroid computation. First, in the assignment step, the algorithm updates the cluster membership by aligning each sensor data stream to all the computed centroids. Each sensor data stream will be assigned to the cluster with the closest centroid. In this step, the k-shape relies on the distance measure of Equation (5) to compare the sensor data stream with the centroids. Second, in the refinement step, the cluster centroids are recalculated whenever new sensor data streams join the cluster. The k-shape method iterates the calculation of the centroid until the cluster membership does not change. Similar to the clustering method k-means, k-shape requires the specification of the number of clusters. However, the correlation between the sensors differs over time, and it is not easy to guess the number of clusters and assign a fixed value. Therefore, in this paper, we use the elbow method to determine the number of clusters.

2) STEP 3-B: CORRELATIONS ANALYSIS

One of the challenges of anomaly detection is reducing the number of false alerts. One essential aspect to consider when developing appropriate sensor anomaly detection is the time span covered by the anomaly itself. Anomalies can span over long periods or short ones. This section introduces the proposed method for an abnormal node detection method that combines a two-phase analysis in ways that can be used to reduce false alerts and detect various attack patterns. It combines median absolute deviation-based analysis to identify short-term anomalies and multicriteria cross-correlation-based analysis to identify and validate both short- and long-term anomalies. This combined detection method aims at mitigating the drawbacks that each of these two methods would have when used separately while making the most of each one's strength.

a: FIRST ANALYSIS PHASE

The first analysis phase provides the first line of the abnormal node detection process to detect short-term anomalies. It uses temporal correlation to extract the temporal changes in the sensor data. Figure 6 illustrates the workflow of the first analysis phase.

The method is based on the MAD measure. In this paper, we use the median MAD instead of the simple mean to avoid assuming that sensor streams are normally distributed. In addition, MAD, which is similar to the mean, computes the median over the absolute deviations from the median but is more robust to point and short-term anomalies. If the collected sensor data streams output a certain percentage of anomalous points with a significant deviation from the median, the MAD will flag these points as anomalous measurements. However, the MAD measure is immune to the length of the sensor data stream. Moreover, the sensor data streams are nonstationary time series that change over time. Therefore, using the MAD as a stand-alone method to detect anomalies may not be effective, as it relies only on a stable median to detect anomalies. Thus, it may mistake nonstationary behavior as misbehavior. Hence, an appropriate time window for the temporal segmentation of sensor streams may be valuable. The aim is to conduct a windowed TC by splitting the sensor data into hourly, daily, weekly, and monthly timespans, and the MAD-based analysis should be monitored at the specified window (Figure 6: 1). Indeed, if we set the window size to a short period, such as one minute, two minutes, or five minutes, we will qualitatively obtain the same results, approximately corresponding to patterns on the minute scale. Of course, at widely different time scales, the sensor correlation may be different. Thus, it is desirable to plan the TC at multiple scales, e.g., hour, day, and month (Figure 6: 2). The system calculates the MAD (periodical check) after each specified TC (Figure 6: 3).

$$MAD_{S_i} = k\ med_i \left|O(S_i) - med_i\ O(S_i)\right|, \tag{6}$$

where $O(S_i)$ is the sensor's collected measurement within the window, $med_i\ O(S_i)$ is the median in the measurements across the sensor data stream, and $k = 1.482$ is the factor scale linked to the assumption of normality of the data, disregarding the abnormality induced by anomalies. The initial window size $initSize$ is given by the user; next, the window size is automatically adjusted based on the concept of MAD-based anomaly detection. The TC determines the boundary point of the sliding window, and $TC_{init} = intiSize$.

As shown in Figure 6, if there is no deviation from each individual sensor data stream (Figure 6: 4), the median indicates that the streams are normal, and the window size will continually increase $w + w'$ until the next planned TC (Figure 6: 5.a). The TC determines the boundary point of the sliding window. On the other hand, if the collected sensor data streams output a certain percentage of anomalous points,

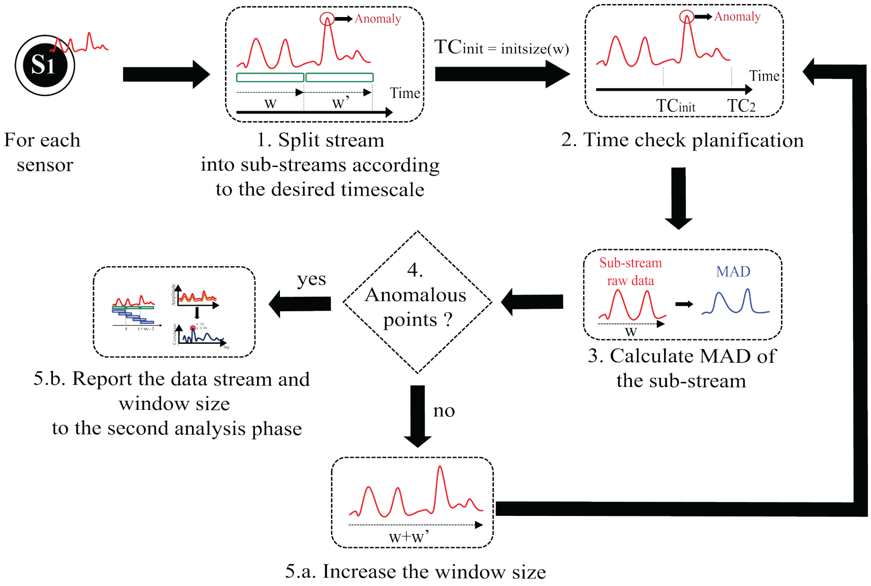

S1
For each sensor
Anomaly
Time
w
w'
1. Split stream into sub-streams according to the desired timescale
TCinit = initsize(w)
Anomaly
Time
TCinit
TC2
2. Time check planification
Sub-stream raw data
MAD
w
3. Calculate MAD of the sub-stream
4. Anomalous points ?
yes
5.b. Report the data stream and window size to the second analysis phase
no
w+w'
5.a. Increase the window size

FIGURE 6. First analysis phase using median absolute deviation.

the MAD will flag these points as anomalous measurements, and all the sensor data streams and the window size will be passed to the second analysis phase (Figure 6: 5.b). When an anomaly is detected, the sensor data stream before the TC is completely removed from the window, and a new w' is formed. Therefore, the TC is moved to the new point where the anomaly is detected. For the mean and standard deviation, it is necessary to define a level of decision. Thus, we must define the rejection criterion of a measurement. $O(S_i, t) = O(S_i, t)_{anomalous}$ when

$$\frac{|O(S_i, t) - med_i \cdot O(S_i, t)|}{MAD_{S_i}} > threshold, \qquad (7)$$

where $O(S_i, t)_{anomalous}$ is the suspicious measurement and *threshold* is the predefined cutoff. In determining how strict the threshold should be, one study [31] proposes values of 3 (very conservative), 2.5 (moderately conservative), or even 2 (poorly conservative). In this paper, we choose a threshold of 2.5 as a reasonable choice.

The first detection method does not aim at high accuracy because it relies on individual sensor measurements and does not exploit spatial correlation data between the sensors. Thus, it is difficult to detect long-term anomalies at the first analysis phase, so detecting long-term anomalies is performed in the second analysis phase.

b: SECOND ANALYSIS PHASE

The detection method takes advantage of correlation in the measurement observation between SNs to increase the detection accuracy. Figure 7 illustrates the workflow of the second analysis phase. In this paper, we propose another practical choice to identify the cross-correlation between two sensor data streams to detect both long- and short-term anomalies. The first step toward capturing the correlation between the sensor data streams over time is to segmentize each sensor data stream to a specific duration of time by using two time window types: the tumbling window and the sliding window (Figure 7: 1).

(a) **Tumbling window:** This is a fixed window that does not overlap, and it is aligned to the next epoch by moving for as long as the window size. The sensor measurements are exclusive for each window. With this method, we can obtain the cross-correlation between the sensor streams from one epoch to another. Thus, we can capture the change in long-term scale correlation.

(b) **Sliding window:** The sliding window, on the other hand, slides ahead one time period across time. New sensor measurements are gradually added at the front, and the older sensor measurements become invisible as the window slides ahead. Note that compared with tumbling windows, the sliding windows overlap between successive epochs. We can obtain the cross-correlation between the sensor streams from one epoch to another with this method. Thus, this will allow us to capture the change in short-term scale correlation. We segment the sensor data stream into a sliding window of p samples with overlapping samples. For instance, as shown in Figure 8, in this paper, we consider 50% overlaps.

Both types of windows move across the sensor data stream, splitting the data into finite subsequences. Overall, the two methods described will help us capture the changes between two sensor data streams over time, making anomaly detection even more accurate. In short, if the length of the time series is long enough and when we are dealing with long-term anomalies, the obtained moving cross-correlation score with a tumbling window may perform better. However, if the length is short and we are dealing with short-term anomalies,

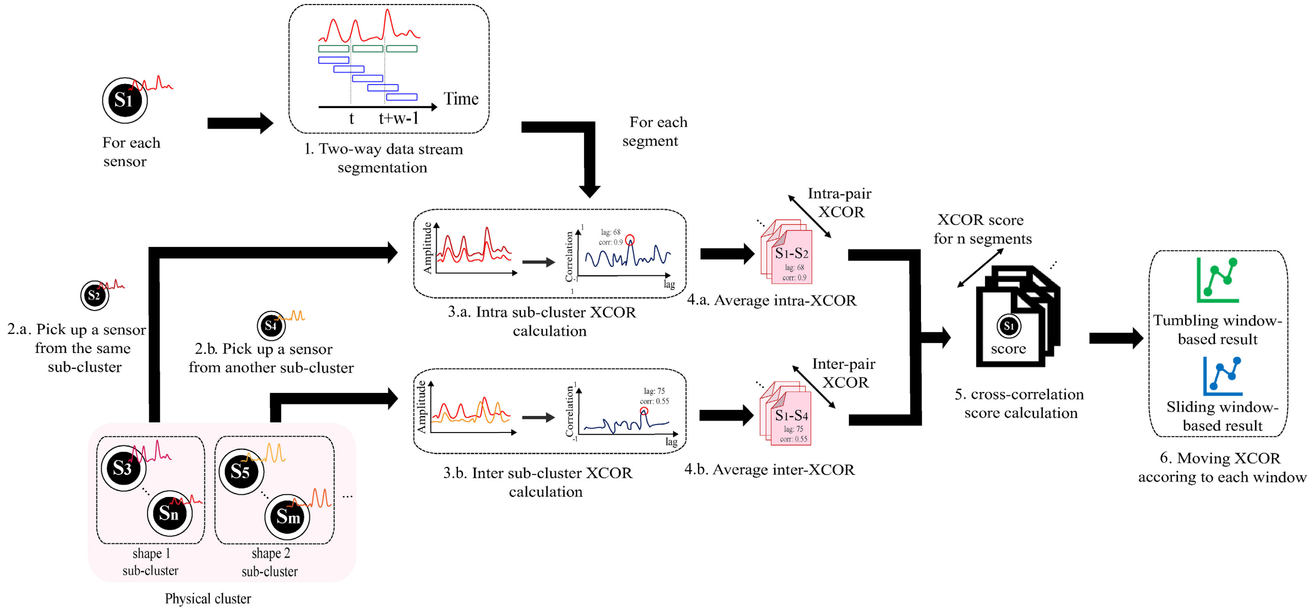

S1
For each sensor
Time
t
t+w-1
1. Two-way data stream segmentation
For each segment
S2
2.a. Pick up a sensor from the same sub-cluster
S4
2.b. Pick up a sensor from another sub-cluster
Amplitude
Correlation
lag
lag: 68
corr: 0.9
3.a. Intra sub-cluster XCOR calculation
lag: 75
corr: 0.55
3.b. Inter sub-cluster XCOR calculation
Intra-pair XCOR
S1-S2
4.a. Average intra-XCOR
Inter-pair XCOR
S1-S4
4.b. Average inter-XCOR
XCOR score for n segments
S1
score
5. cross-correlation score calculation
Tumbling window-based result
Sliding window-based result
6. Moving XCOR accoring to each window
S3
Sn
S5
Sm
shape 1 sub-cluster
shape 2 sub-cluster
Physical cluster

FIGURE 7. Second analysis phase.

then the obtained moving cross-correlation score with a sliding window may be a better choice. After completing the data stream segmentation according to each time window, we proceed to the cross-correlation calculation step. For each sensor data stream segment, the detection system proceeds to the multicriteria cross-correlation calculation as part of the decision-making process to identify anomalies. Heterogenous sensor data tend to exhibit a strong correlation in both space and time that can be used to enhance the abnormal node detection problem in WSNs. Corruption in these correlations is very likely to be affected by the presence of anomalies. To estimate the correlation between sensor data streams, one promising idea is to use the concept of lag correlation given by the CCF time [32]. Two sensor streams have a lag correlation of l if they look very similar when one is delayed by l time ticks. The Pearson formula is adopted as the criterion for the lag correlation:

$$Xcorr(O(\vec{S}_i), O(\vec{S}_j), l) = \frac{\sum_{t=l+1}^{n}(O(S_i, t) - \overline{O(\vec{S}_i)})}{\sqrt{\sum_{t=l+1}^{n}(O(S_i, t) - \overline{O(\vec{S}_i)})^2}} \times \frac{\sum_{t=l+1}^{n} O(S_j, t-l) - \overline{O(\vec{S}_j)}}{\sqrt{\sum_{t=1}^{n-l}(O(S_j, t) - \overline{O(\vec{S}_j)})^2}} \quad (8)$$

where $\overline{O(\vec{S}_i)} = \frac{1}{n-1}\sum_{t=l+1}^{n} O(S_i, t)$ and $\overline{O(\vec{S}_j)} = \frac{1}{n-1}\sum_{t=1}^{n} -O(S_j, t)$. Here, $Xcorr(O(\vec{S}_i), O(\vec{S}_j), l)$ represents the correlation coefficient when one stream is delayed by l. Based on Equation (8), we can now define how we evaluate the correlations between homogeneous sensor streams. The homogeneous intracluster correlation is defined formally as follows. Given two numerical same-type sensor streams located in the same cluster and observed at the same time t, let $Cp(S_i) = Cp(S_j)$ and $T(S_i) = T(S_j)$. For all S_i and S_j with $(i \neq j)$, S_i and S_j sensor streams are considered correlated if:

(a) the score $(|Xcorr(O(\vec{S}_i), O(\vec{S}_j), l)|)$ between $O(S_i, t)$ and $O(S_j, t-1)$ is actually a local maximum, and
(b) is the earliest such maximum if additional maxima exist.

The reason for the second condition relates to the case where the two sequences are periodic with the same period T (for real sequences, this will be daily or yearly).

We will then obtain more than one local maximum (e.g., l, $l + T$, and $l + 2 \times T$). Clearly, the earliest lag is the most important to consider. In this paper, when extracting the hidden correlations between the sensor data streams and deciding whether the sensors are abnormal, multicriteria *Xcorr* need to be accounted for. Using Equation (8), we proceed to extract the pairwise *Xcorr* with all other sensors within the same cluster for each sensor. For this, we compute two kinds of multicriteria *Xcorr*, namely, intra-subcluster *Xcorr* and inter-subcluster *Xcorr*. As explained in subsection IV. C, each cluster is divided into logical subclusters of sensors with similar structure patterns. For each sensor, we proceeded to extract the intrapairwise *Xcorr* (Figure 7: 3.a) with all other sensors within the same logical subcluster (Figure 7: 2.a).

Once all the intrapairwise *Xcorr* are extracted, we calculate the average intra-subcluster *Xcorr* of each sensor data stream to other sensors (Figure 7: 4.a). The average intra-subcluster *Xcorr* is calculated as follows:

$$\overline{Corr(S_i)}_{intra} = \frac{1}{size(Cl(S_i))}\sum_{i \neq j} Xcorr(O(\vec{S}_i), O(\vec{S}_j))_{intra}, \quad (9)$$

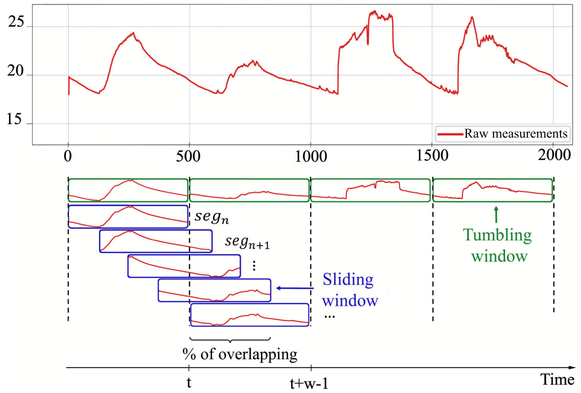


FIGURE 8. Sensor data stream segmentation with two time-based windows.

where $size(Cl(S_i))$ is the size of the logical cluster Cl where S_i belongs. $Xcorr(O(\vec{S_i}), O(\vec{S_j}))_{intra}$ is the correlation between S_i and S_j. Here, S_j runs over all sequences in the set except for S_i itself. $\overline{Corr(S_i)}_{intra}$ is the intra-average correlation. Thus, each sensor calculates the intra-average of the correlation.

This also applies to the second criteria of $Xcorr$. The inter-pairwise $Xcorr$ (Figure 7: 3.b) is extracted for each sensor with all other sensors belonging to other logical subclusters (Figure 7: 2.b). Once all the interpairwise $Xcorr$ are extracted, we calculate the average inter-$Xcorr$ of each sensor data stream to other sensors (Figure 7: 4.b). The average inter-$Xcorr$ is calculated as follows:

$$\overline{Corr(S_i)}_{inter} = \frac{1}{size(Cp(S_i))} \sum_{i \neq j} Xcorr(O(\vec{S_i}), O(\vec{S_j}))_{inter}, \tag{10}$$

where $size(Cp(S_i))$ is the size of the physical cluster Cp where S_i belongs. $Xcorr(O(\vec{S_i}), O(\vec{S_j}))_{inter}$ is the correlation between S_i and S_j. Here, S_j runs over all sequences in the set except for S_i itself. $\overline{Corr(S_i)}_{inter}$ is the interaverage correlation. Thus, each sensor calculates the interaverage of the correlation.

Afterward, the weighted average of the observed average of both $\overline{Corr(S_i)}_{intra}$ and $\overline{Corr(S_i)}_{inter}$ is calculated to obtain S_i's correlation score for the n analyzed segments (Figure 7: 5.):

$$\overline{Corr(S_i)} = w_{intra} \cdot \overline{Corr(S_i)}_{intra} + w_{inter} \cdot \overline{Corr(S_i)}_{inter}, \tag{11}$$

where $w_{intra} = \frac{size(Cl(S_i))}{size(Cp(S_i))}$ and $w_{inter} = 1 - w_{intra}$ are the assigned weights to the intra-average correlation and interaverage correlation, respectively.

The motivation for weighting the intra-average correlation and interaverage correlation is to give more importance to the intracorrelation between the sensors while also taking into consideration the intercorrelation between them. Imagine that we only consider inter-average correlation or assign a very large weight (for example, 0.9) to interaverage correlation; the final sensor correlation score will be similar to the interaverage correlation. Thus, to assign the appropriate combination weight for each average correlation, we refer to the sizes of both the physical cluster and the logical cluster of each sensor S_i.

TABLE 3. MVA correlation extraction.

Rule	Node value	ST-Cr	HSAV	Decision
1	Normal	Correlated	At least one is normal	Normal
2	Normal	Uncorrelated	All normal	Abnormal
3	Abnormal	Correlated	At least one is abnormal	Event
4	Abnormal	Uncorrelated	All normal	Abnormal

In the experimental section, we show that combining relevant observations from both average correlations by discriminative weighting provides a possible way to improve the detection accuracy. Once all the segments of each time window have been analyzed, we finally obtain two types of moving correlation scores: the tumbling-window-based moving correlation score and the sliding window-based moving correlation score (Figure 7: 6).

3) STEP 3-C: SENSOR SIMILARITY DEGREE AND STATE IDENTIFICATION

The final step involves collecting all the extracted features of each sensor to calculate its similarity degree with other sensors located in the same cluster (Figure 3).

1) Let $f_1(\vec{S_i})$ be S_i's z-normalized time series.
2) Let $f_2(\vec{S_i})$ be S_i's moving MAD time series.
3) Let $f_3(\vec{S_i})$ be S_i's tumbling-window-based moving $Xcorr$ time series.
4) Finally, let $f_4(\vec{S_i})$ be S_i's sliding-window-based moving $Xcorr$ time series.

When all these extracted features are integrated, valuable insight can be obtained, which makes abnormal node detection even more accurate. Therefore, the four observed features of the time series are fed into the shape-based clustering process. Then, we calculate the degree of similarities between the different SNs for each extracted feature f_o, $o \in [1, 4]$. Let $f_o(\vec{S_i}) = \{f_o(S_i, 1), \ldots, f_o(S_i, m)\}$ and $f_o(\vec{S_j}) = \{f_o(S_j, 1), \ldots, f_o(S_j, m)\}$ be two extracted time-series features of length m of sensors S_i and S_j, respectively ($i \neq j$), and $T(S_i) = T(S_j)$ and $Cp(S_i) = Cp(S_j)$. To determine the similarity between $f_o(\vec{S_i})$ and $f_o(\vec{S_j})$, we use Equation (5). If the two time series share similar characteristic patterns, then they will be classified in the same cluster. Let $Wt_{i,j}$ be the weight that represents the number of times when S_i and S_j share similar characteristic patterns for each extracted feature f_o. The similarity degree of sensor S_i is calculated as follows:

$$SimDeg(S_i) = \frac{\sum_{i \neq j} Wt_{i,j}}{(sizeCp(S_i) - 1) \cdot nbfeature}, \tag{12}$$

where it takes a value between 0 and 1, with 0 indicating no similarity between the sensors. For the analysis of anomalies,

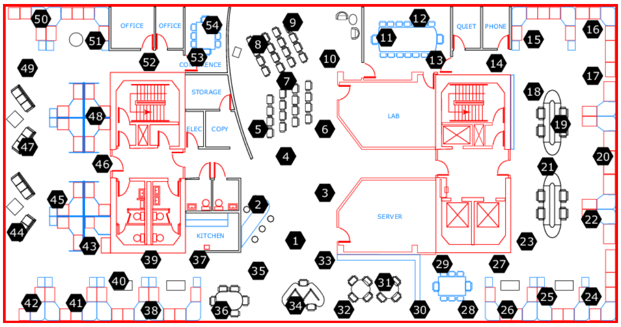


FIGURE 9. Sensors deployed in the Intel berkeley research lab.

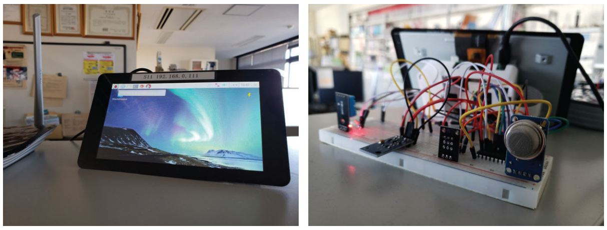


FIGURE 10. Raspberry Pi 2 equipped with four types of sensors.

the correlation state of sensor node S_i is considered to be uncorrelated if sensor similarity degree $SimDeg(S_i)$ is less than *threshold* $= 0.5$.

D. STEP 4: MULTIVARIATE-ATTRIBUTE CORRELATION EXTRACTION

Homogeneously sensed data usually contain both time and space information. However, the data generated by heterogeneous sensors are not independent, and we can obtain additional valuable information that may lead to better insights into the monitored environment. We may call this the "observed MVA." All multivariate attributes sensed directly by heterogeneous sensors and analyzed in the previous steps will be used in forming a final decision; however, they may not be sufficient to guarantee an accurate decision. In typical situations, when there is no interference from events or abnormal nodes, heterogeneous sensor streams located in the same cluster tend to be correlated, whereas the occurrence of events usually appears in specific observations.

For example, a particular change in sensor data caused by an event will exhibit temporal and spatial correlation. This change will continue for a while after the event occurs. Moreover, if the event occurs in a specific cluster, then heterogeneous sensors located in the same cluster will show a high degree of correlation, unlike sensors located in other clusters. For example, in a fire detection system, light intensity, temperature, and smoke density are all necessary elements of the information used to indicate that a fire has been detected. By taking into account such heterogeneous observations, we can be confident about the accuracy of the event detection without it being mistaken for abnormal node behavior. The occurrence of abnormal nodes tends to involve uncorrelated singular nodes within a cluster. Based on the observed MVA correlations, a set of a few rules can be devised to check if an abnormal sensor exists while also identifying actual events.

TABLE 4. List of abbreviations and acronyms.

IoT	Internet of things
WSNs	Wireless sensor networks
BS	Base station
CH	Cluster head
SN	Sensor node
FDIA	False data injection attack
ST	Spatiotemporal
MVA	Multivariate-attributes
SBD	Shape-based distance
CCF	Cross-correlation function
Xcorr	Cross-correlation
MAD	Median absolute deviation
TC	Time check

TABLE 5. Symbols and notations.

Symbol	Notation
S_i	Sensor i
$T(S_i)$	Type of sensor i
$L(S_i)$	Geographical location of sensor i
$Cp(S_i)$	Physical cluster of sensor i
$size(Cp(S_i))$	Size of the physical cluster Cp where sensor i belongs
$Cl(S_i)$	Logical cluster of sensor i
$size(Cl(S_i))$	Size of the logical cluster Cl where sensor i belongs
$O(\vec{S_i})$	Data stream of sensor i
$O(S_i, t)$	Measurement of sensor i at time t
N	Number of sensor nodes
m	Length of a sensor data stream
$f_o(\vec{S_i})$	Time-series feature f_o of sensor i

Table 3 defines the set of rules. Each rule has three inputs (antecedents) called the node value, the ST correlation results obtained from the previous step (ST-Cr), and the heterogeneous sensor average (HSAV), and one output (the consequent) is termed the decision. The node value and ST-Cr are properties of the node to be analyzed, and HSAV is a vector that represents the average values of the heterogeneous sensor data $\{H_1, H_2, \ldots\}$, where H_i is the average value of all the same-type sensors located in cluster i. An average value is considered abnormal if the value is outside a predefined range. The output, i.e., the decision, will indicate whether the sensor node is normal, abnormal, or an event is detected. If abnormal nodes exist, then such nodes are reported.

The final decision is based on these predefined rules. For each rule, if all the antecedents are satisfied, then the consequent is true.

TABLE 6. Experiment environment.

	Raspberry Pi 2-based	**Raspberry Pi 3-based**	**Base station**
CPU	ARM Cortex-A7 Quad Core 900 MHz	ARM Cortex-A53 Quad Core 1.2 GHz	AMD Opteron 4184 (6 cores / 2.8 GHz) x 2
RAM	1 GB	1 GB	32 GB
OS	Debian 9.3 (Raspbian Stretch)	Debian 9.3 (Raspbian Stretch)	Ubuntu 17.10
Python	2.7.13	2.7.13	3.7.4

TABLE 7. Sensor types.

Sensor	Type	Normal values
Temperature	DS18B20 temperature sensor	[15°C, 30°C]
Humidity	DHT11 humidity sensor	[30%, 50%]
Smoke density	MQ-2 smoke sensor	[0 ppm, 0.2 ppm%]
Light intensity	BH1750 digital light sensor	[0 Lux, 2000 Lux]

V. EXPERIMENTAL SETUP

This section describes the datasets used to evaluate our proposed approach and the details of the sensor network that we have implemented, including the deployment setting and the experimental scenario design.

A. DATA ACQUISITION

To show that our proposed approach is applicable to real-world WSNs deployed with heterogeneous sensors, we use two datasets that have different types of sensor deployment. The summary of the datasets is shown in Table 8. The first dataset is the publicly available Intel Berkeley Research Lab dataset [33]; the second dataset is the data collected from our deployable WSN in our laboratory.

1) INTEL BERKELEY RESEARCH LAB (INTEL LAB)

In this dataset, the data were collected from 54 SNs deployed in the Intel Berkeley Research Lab between February 28 and April 5, 2004 [33]. To effectively monitor the whole lab environment, 54 sensors are unevenly distributed in different locations in the research lab. Mica2Dot sensors with weatherboards are used to collect time-stamped topology information, along with temperature (in degrees Celsius), humidity (temperature-corrected relative humidity ranging from 0–100%), light (Lux) (a value of 1 Lux corresponds to moonlight, 400 Lux to a bright office, and 100,00 Lux to full sunlight), and voltage values (in volts ranging from 2 to 3). The batteries, in this case, were lithium-ion cells that maintained a reasonably constant voltage over their lifetime. A new reading was collected almost every 31 seconds. In total, 2.3 million readings were collected from these sensors. The sensors were dispersed in the lab, as shown in Figure 9.

2) OUR LAB DATASET (YOKOTA LAB)

In addition to the Intel Lab dataset, we also collected sensor data streams from 27 SNs in our laboratory between January 24 and July 25, 2018. The real-world sensor data were collected periodically while performing our usual daily activities. The SNs were deployed using the Raspberry Pi 2 and 3 Model B microcontroller platforms, as we consider the Raspberry Pi to be the best IoT hardware platform in terms of performance and flexibility (see Figure 10). Each physical sensor node is equipped with one temperature sensor module, one humidity sensor, one smoke-density sensor, and one digital light-intensity sensor, yielding a total of 64 sensors. The technical characteristics of the Raspberry Pi platforms, sensors, and server used in our experimental setting are described in Figure 9, Table 6, and Table 7, respectively. As shown in Figure 11, the SNs were divided into five clusters separated from each other and with different environmental conditions. Two clusters comprised five SNs each and were located in our laboratory room. The third consisted of six physical nodes located in a kitchen corner and exposed to sunlight, the fourth consisted of six physical nodes located in a seminar room and the fifth consisted of five physical nodes located in a server room. Each sensor transmits data approximately every 1 min, giving a total of 20.9 million readings.

B. DATA PREPROCESSING

Three main steps must be performed to prepare the dataset for the evaluation: cleaning the raw sensor data, injecting false sensor data, and physically separating the SNs into clusters. Cleaning the data is necessary to ensure that the proposed abnormal node detection is only executed on known FDIAs, allowing for consistent evaluations. After that, new false sensor data may be injected. The clustering process is also considered a necessary process in this paper to capture the sensor data correlation adequately. In the following subsections, we explain the three main steps in more detail.

1) DATA CLEANING

The main challenge in cleaning the dataset is the fact that the process cannot be fully automated, as no general appropriate method of detecting faulty sensor data exists. Although several automated preprocessing techniques have been proposed for sensor data, this paper considers a manual technique for preprocessing the two datasets. To use the Intel Berkeley Research Lab dataset, we faced some challenges during preprocessing. The main issue encountered was the data related to the notion of time variation (i.e., epoch). Indeed, the usage of the epoch is necessary to build a baseline that works on sensor data streams such as our collected dataset or Intel dataset. However, for the case of the Intel dataset and even our dataset, the notion of the epoch is loosely defined.

Indeed, even though SNs are commanded to collect a new reading in every defined epoch, the fact of having multiple values or missing values for different epochs cannot be escaped (because of failures or communication problems). For the WSNs deployed at the Intel Lab, the reasons behind the failures were communication problems and the sensor battery condition. In addition, we found that readings of sensor node five in the Intel Lab data were not recorded. Consequently, it was removed from the dataset.

With regard to our deployed WSNs, some SNs had missing data for different epochs because of SD card corruption. The concept of the epoch is necessary to establish a baseline while working on sensor streams such as our collected dataset or the Intel Lab dataset. However, because of the sensor constraints, we found that the epoch was not strongly defined in either dataset. Thus, we needed to standardize the concept of epoch and set it to a well-defined size. To unify the size, we split the readings into epochs of two minutes each.

TABLE 8. Datasets.

	Number of sensors	Sensor types	Number of clusters	Period	Dataset size
Yokota Lab	27	Temperature, humidity, light intensity, and smoke	5 (each include 5, or 6 sensor nodes)	Between January 24 and July 25, 2018	20.9 million readings (new reading every 2 min)
Intel Lab	54	Temperature, humidity, and light intensity	11 (each include 5 or 4 sensor nodes)	Between February 28 and April 5, 2004	2.3 million readings (new reading every 2 min)

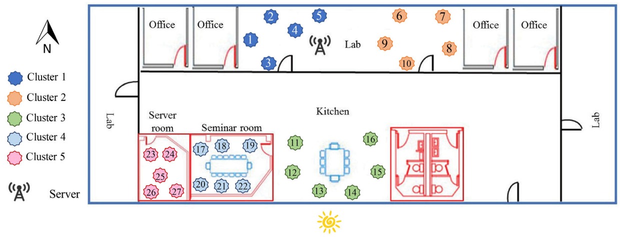


FIGURE 11. **Heterogeneous SNs deployed in our laboratory.**

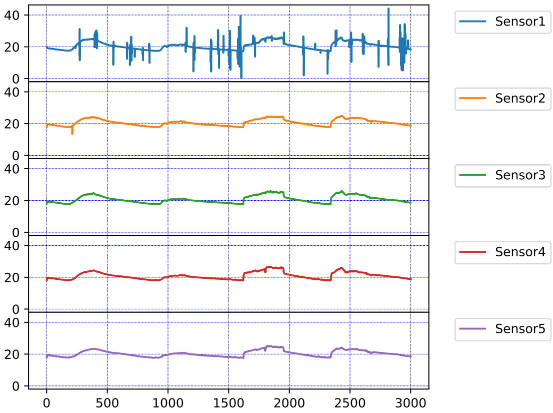


FIGURE 12. **An example of collected temperature sensor data in one cluster with sensor 1 being an abnormal node.**

The value of the last obtained reading was used to substitute for each missing sensed value in an epoch. Moreover, if a sensor had more than one reading during the epoch, we took the average of these measurements.

2) FALSE SENSOR DATA INJECTION

Given the lack of sensor datasets with malicious data for WSNs and the need to test our approach's accuracy, we propose an FDIA model to create an attack strategy. An attacker's goal in the context of FDIA is to evoke or hide events without triggering the detection alarm. The primary challenge is to maintain a balance between the outcome of the attack and the risk of being detected. In [18], the proposed attack models are unsophisticated and not comprehensive enough to support the claims in the paper. We only considered three trivial cases where the attacker deliberately either randomly changes a sensor reading or selects the minimum or the maximum possible value. We carefully chose the type of attacked sensor, insertion time, and attack period. With such a proposed attack strategy, it is impossible to guarantee a variety of attack patterns, which results in uninteresting attack outcomes that are easy to detect. WSNs are subject to various threats where we cannot simply anticipate the attacker's actual attention.

To tackle this issue, in this paper, we propose an attack strategy to generate a malicious dataset from the original sensor data, which allowed us to test the detection algorithm and evaluate its performance against different threat severity levels. We create evaluation data, including FDIA and missing data based on the initially collected dataset. Let the occurrence probability of missing data depend on the exponential distribution.

$$f(e) = \frac{1}{\epsilon} exp(-\frac{e}{\epsilon}), \tag{13}$$

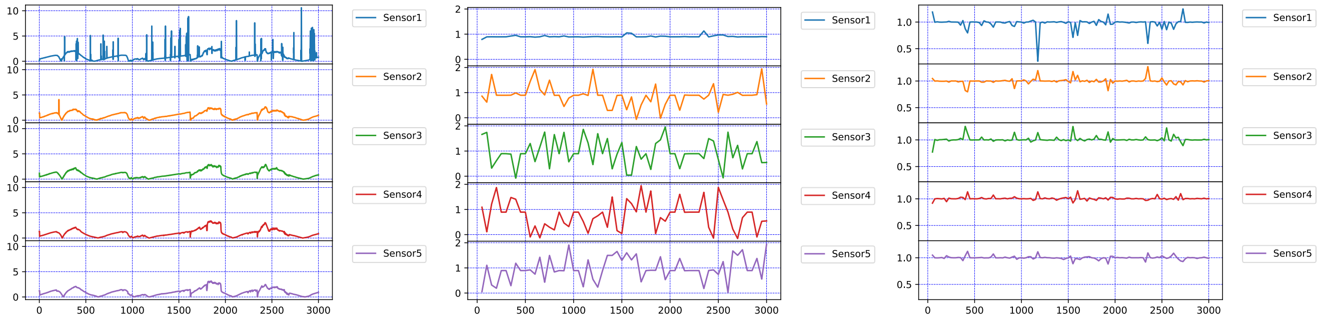
where $(500 \leq \epsilon \leq 1000)$. In addition, we defined nine types of false data and one incidence of missing data. The false injected data difference between the real data and the evaluation data depends on a Gaussian distribution.

$$f(e) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{-\frac{(e - O(S_i, t))^2}{2\sigma^2}\right\}, \tag{14}$$

where $(0 \leq \sigma^2 \leq 10)$. We referred to both Equations (13) and (14) and injected false sensor data readings into the initially collected sensor data. The sensor type, FDIA type, and insertion time were chosen randomly. With such an FDIA strategy, the attack can be very stealthy and deceive the detection mechanism without being easily detected. Moreover, we can expect to have various attack patterns that span over a long or short period (Figure 2). In this paper, we assume that an attacker cannot compromise many SNs, as it is challenging and difficult to achieve without being detected. Thus, we consider one abnormal sensor node in each cluster at a time in our experiment. Figure 12 illustrates an example of collected temperature data from the Yokota Lab dataset, with sensor one being under the proposed FDIA. Figure 13 illustrates an example of the extracted features of each sensor after conducting the ST correlation extraction explained in Section IV.C. The results show that the shape of the extracted time-series features of sensor one significantly differs from its neighbors. The correlation state of sensor one will be considered uncorrelated. Thus, sensor one will be identified as an abnormal node.

3) PHYSICAL CLUSTERING

The clustering method we used is simple k-means clustering. We used geographic coordination (i.e., Euclidean distance) as clustering parameters. Other proposed clustering techniques can also be applied.



(a) Moving MAD time series (b) Tumbling-window-based $Xcorr$ time series (c) Sliding-window-based $Xcorr$ time series

FIGURE 13. An example of all the extracted features of neighboring temperature sensors in one cluster.

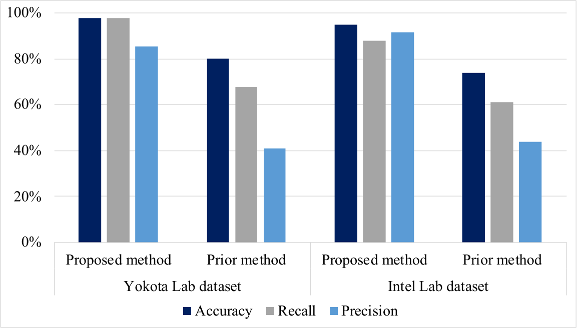


FIGURE 14. Detection accuracy of the two datasets.

VI. EVALUATION

Each conducted experiment was repeated five times, and we took the average results.

A. DETECTION ACCURACY

To evaluate our proposed method in terms of abnormal node detection, three performance metrics were used, namely, accuracy, precision, and recall. We used precision and recall to quantify the detection accuracy. Accuracy is the degree to which the detection results confirm the actual values. Precision is the percentage of the actual abnormal SNs among the identified SNs. Recall is the percentage of the identified abnormal nodes among the actual abnormal sensors. Figure 14 shows the evaluation results of the two datasets that measure the extent to which our proposed method and the prior method proposed in [18] detect abnormal SNs.

Even though the environmental conditions for each cluster in the two datasets were different, the proposed method in this paper achieved high detection accuracy with a low false-positive rate for the task of analyzing the sensor readings to determine whether the SNs were behaving normally or had been exposed to FDIAs. The results show that our new method achieved an average accuracy of 96.50%, an average precision of 88.69%, and a recall rate of 93.00%. Moreover, the proposed method in this paper achieved better detection results than the detection results achieved using the prior method proposed in [18]. The results show that our proposed method achieved an average accuracy improvement of over 22.25%, an average precision improvement of over 70.53%, and a recall rate improvement of over 36.20%. Therefore, we conclude that the proposed method in this paper detects abnormal nodes with high accuracy.

Relatively, the Intel Lab dataset showed slightly lower precision and recall than the Yokota Lab dataset; the reason is shown in Figure 15 and Figure 16. We calculated the correlation degree, which indicates how much correlation exists between each pair of sensors. In Figure 15 and Figure 16, we show the overall Yokota Lab temperature and Intel Lab temperature correlation between all sensor pairs using a heatmap. To quantify the degree of correlation, we calculated the rate of pairs of sensors with high correlation degrees regarding the total number of sensors. In the Yokota Lab dataset, the rate of pairs of temperature sensors with high correlation degrees was 76.92%, while in the Intel Lab dataset, the rate of pairs of temperature sensors with high correlation degrees was 61.59% and had more deployed SNs. This explains why the number of deployed SNs and the correlation degree are not directly proportional. Alternatively, the detection accuracy was more dependent on the correlation degree between the SNs in each dataset.

1) IMPACT OF MULTIABNORMAL NODE

In this paper, we considered one malicious sensor in each cluster during our experiment at a time, but multiple malicious sensors may co-occur in WSNs in real-world situations. Note that such sophisticated attacks require a sound and well-planned strategy and thus are difficult to automate. However, even though such a case is unlikely to occur, we injected false data in one to three sensors in a cluster to generate multiple malicious sensors simultaneously and examine the result. The evaluation results are shown in Figure 17. The average accuracy, recall, and precision for detecting multiple malicious sensors (two and three abnormal nodes) per cluster were 83.10%, 64.20%, and 76.34%, respectively, within a reasonable range. The purpose of the experiments is to show that even with a sophisticated multiabnormal node strategy, our proposed method can detect the attack and correctly characterize the abnormal sensors under certain conditions.

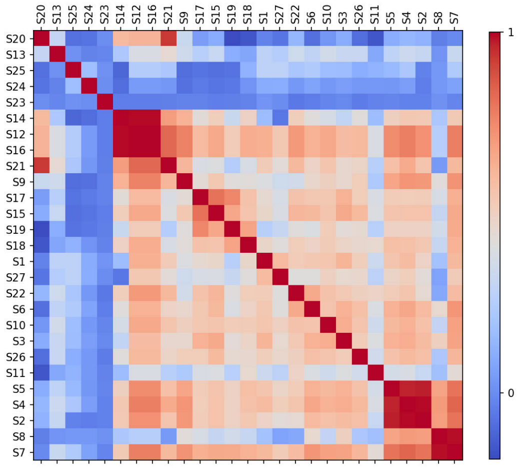


FIGURE 15. Sensor correlation degree in the Yokota lab dataset.

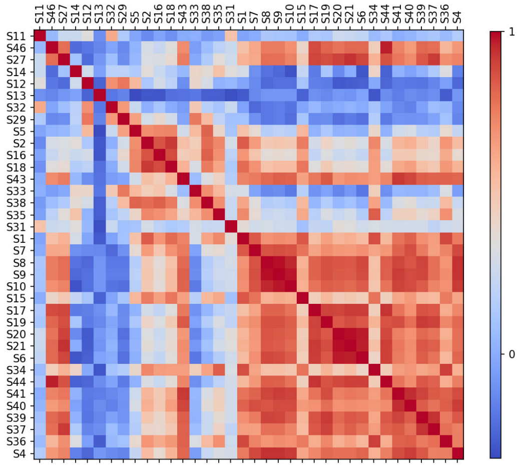


FIGURE 16. Sensor correlation degree in the intel lab dataset.

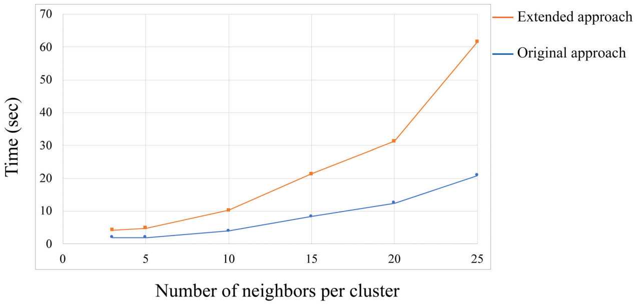


FIGURE 17. Effect of the number of neighbors per cluster on the computation time.

B. COMPUTATION TIME AND IMPACT OF DIFFERENT PARAMETERS

In addition to evaluating the detection accuracy, we also measured the effect of the number of neighbors per cluster on the computation time. We used the Intel dataset for this experiment, as it has more SNs than the Yokota Lab dataset.

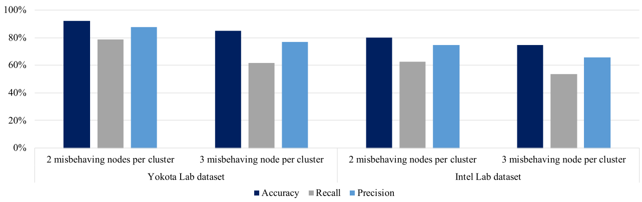


FIGURE 18. Evaluation results under different attack intensities.

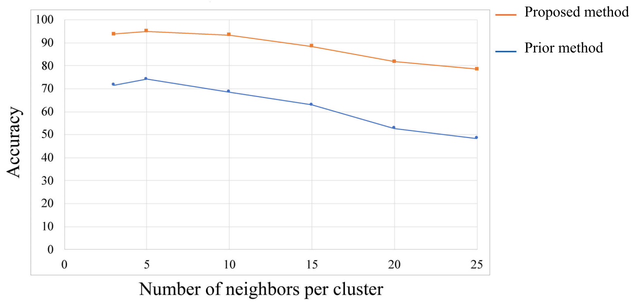


FIGURE 19. Effect of the number of neighbors per cluster on the detection accuracy.

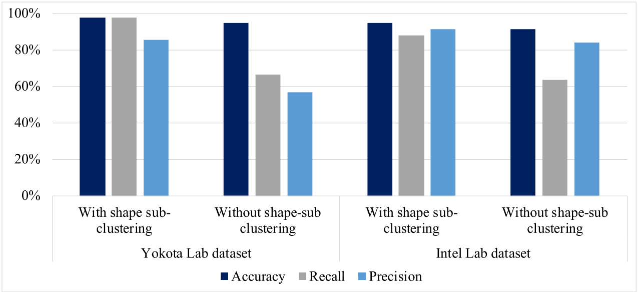


FIGURE 20. The advantage of using multicriteria Xcorr.

The average computation time for our method is the time to analyze the newly collected sensor streams (sampling once per 100 rounds) and execute the shape-based subclustering module, abnormal node detection module, and sensor similarity module. The evaluation results are shown in Figure 18. As shown in the figure, the proposed method in this paper was approximately four times slower than the prior method in [18].

This is because the newly proposed method requires additional processing to enhance the detection accuracy, as shown in Figure 16. The newly proposed method performs 30.29% better than [18] in terms of detection accuracy, and the computation time increases with the increased number of neighbors per cluster.

However, we do not need to group a large number of sensors in one cluster. As shown in Figure 16, the detection accuracy declines with an increased number of neighbors per cluster. The reason for this phenomenon is the effect of the correlation degree between the sensors. If we group a large number of sensors in one cluster, there is no guarantee that they will be under the same dynamic environmental characteristics. The clustering process can reduce the computational complexity of the proposed method and improve the detection accuracy of abnormal SNs. For this reason, we need to have more clusters rather than to group many SNs into a few clusters.

Figure 19 shows that a cluster should include from three to 10 SNs for better detection accuracy. Moreover, Figure 19 shows that compared with the prior method, the detection accuracy proposed method in this paper decreases slightly with an increasing number of clusters. The reason behind this phenomenon is that in this paper, apart from physical clustering, we also perform logical subclustering to guarantee a better extraction for the correlation between the sensors and improve the detection accuracy. Figure 20 shows the evaluation results when considering shape subclustering.

VII. CONCLUSION

This paper explores the challenges faced by environment monitoring applications using WSNs and presents a new approach to address these issues. The proposed method detects abnormal nodes in WSNs by considering both ST and MVA correlations.

The cross-correlation between the sensor data streams is extracted in both space and time using shape-based logical subclustering and a two-phase analysis method. In the first analysis phase, the system uses a variable-size sliding window and the MAD measure. If the collected sensor data streams output a certain percentage of anomalous points, the MAD will flag these points as anomalous measurements, and all the sensor data and the window size will be passed to the second analysis phase. The latter performs both tumbling-window and sliding-window analyses to extract multicriteria cross-correlation measures. Both types of windows move across the sensor data stream, splitting the data into finite subsequences. Thus, it will help us capture the changes between two neighboring sensor data streams over time, making anomaly detection even more accurate. Finally, all the extracted sensor time-series features will be fed to the shape-based clustering to generate a sensor similarity-like graph. The latter reflects the similarity degree of the sensor with the other nodes. The nodes with a low similarity degree below the threshold will be reported as abnormal nodes. The nodes with a low similarity degree below the threshold will be identified as candidate abnormal nodes. Based on the observed MVA correlations, four rules are introduced to check whether the detected candidate abnormal nodes represent actual events. Finally, if abnormal nodes exist, then such nodes are reported. We also propose a new attack strategy to generate malicious datasets from the original sensor data, allowing us to test the detection algorithm and evaluate its performance against different threat severity levels. We create evaluation data, including various FDIA patterns and missing data, based on the initially collected dataset.

Our experiments using two real-world datasets demonstrate that our proposed method detects abnormal nodes with an average accuracy of 96.50%, an average precision of 88.69%, and a recall rate of 93.00%.

Although many studies have reported addressing the abnormal node detection problem in WSNs, it is difficult to compare their performance. As introduced in the previous sections, the design assumptions and the experimental environments are very different. In particular, the lack of a comparable benchmark thwarts a meaningful comparison of the detection results.

REFERENCES

[1] L. Mottola and G. P. Picco, "Programming wireless sensor networks: Fundamental concepts and state of the art," *ACM Comput. Surveys*, vol. 43, no. 3, pp. 1–51, Apr. 2011.

[2] K. Ni, N. Ramanathan, M. N. H. Chehade, L. Balzano, S. Nair, S. Zahedi, E. Kohler, G. Pottie, M. Hansen, and M. Srivastava, "Sensor network data fault types," *ACM Trans. Sensor Netw.*, vol. 5, no. 3, pp. 1–29, May 2009.

[3] K. Kapitanova, E. Hoque, J. A. Stankovic, K. Whitehouse, and S. H. Son, "Being SMART about failures: Assessing repairs in SMART homes," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 51–60.

[4] G. Padmavathi and D. Shanmugapriya, "A survey of attacks, security mechanisms and challenges in wireless sensor networks," *Proc. Int. J. Comput. Sci. Inf. Secur.*, vol. 4, nos. 1–2, pp. 117–125, 2009.

[5] C. Franzen. (2013). *Dick Cheney Had the Wireless Disabled on his Pacemaker to Avoid Risk of Terrorist Tampering*. Accessed: Mar. 22, 2021. [Online]. Available: https://www.theverge.com/2013/10/21/4863872/dick-cheney-pacemaker-wireless-disabled-2007

[6] T. Kavitha and D. Sridharan, "Security vulnerabilities in wireless sensor networks: A survey," *J. Inf. Assurance Secur.*, vol. 5, no. 1, pp. 31–44, 2010.

[7] N. Berjab, C. M. Yu, S. Y. Kuo, and H. Yokota, "Impact analysis for DoS and integrity attacks on IoT systems," in *Proc. 7th Int. Conf. Inf. Syst. Technol.*, 2017, pp. 1–8.

[8] (2018). *Hidden IoT Security Issues Pose a Huge Threat to Your Network*. Accessed: Mar. 22, 2021. [Online]. Available: https://www.arcserve.com/jp/insights/iot-security-issues/

[9] J. Radcliffe. (2011). *Hacking Medical Devices for Fun and Insulin: Breaking the Human SCADA System*. Accessed: Mar. 22, 2021. [Online]. Available: https://media.blackhat.com/bh-us-11/Radcliffe/BH_US_11_Radcliffe_Hacking_Medical_Devices_WP.pdf

[10] B. D. Weinberg, G. R. Milne, Y. G. Andonova, and F. M. Hajjat, "Internet of Things: Convenience vs. privacy and secrecy," *Bus. Horizons*, vol. 58, no. 6, pp. 615–624, Nov. 2015.

[11] M. Xie, S. Han, B. Tian, and S. Parvin, "Anomaly detection in wireless sensor networks: A survey," *J. Netw. Comput. Appl.*, vol. 34, no. 4, pp. 1302–1325, 2011.

[12] I. C. Paschalidis and Y. Chen, "Statistical anomaly detection with sensor networks," *ACM Trans. Sensor Netw.*, vol. 7, no. 2, pp. 1–23, Aug. 2010.

[13] J. Park, R. Ivanov, J. Weimer, M. Pajic, and I. Lee, "Sensor attack detection in the presence of transient faults," in *Proc. ACM/IEEE 6th Int. Conf. Cyber-Phys. Syst.*, Apr. 2015, pp. 1–10.

[14] S. R. Arashloo, J. Kittler, and W. Christmas, "An anomaly detection approach to face spoofing detection: A new formulation and evaluation protocol," *IEEE Access*, vol. 5, pp. 13868–13882, 2017.

[15] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Secur.*, vol. 14, no. 1, pp. 1–13, Jan. 2011.

[16] Y. Mo, E. Garone, A. Casavola, and B. Sinopoli, "False data injection attacks against state estimation in wireless sensor networks," in *Proc. 49th IEEE Conf. Decis. Control*, Dec. 2010, pp. 5967–5972.

[17] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using Kalman filter," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 370–379, Dec. 2014.

[18] N. Berjab, H. H. Le, C.-M. Yu, S.-Y. Kuo, and H. Yokota, "Abnormal-node detection based on spatio-temporal and multivariate-attribute correlation in wireless sensor networks," in *Proc. IEEE 16th Int. Conf. Dependable, Autonomic Secure Comput.*, Aug. 2018, pp. 568–575.

[19] H. H. Bosman, G. Iacca, A. Tejada, H. J. Wörtche, and A. Liotta, "Spatial anomaly detection in sensor networks using neighborhood information," *Inf. Fusion*, vol. 33, pp. 41–56, Jan. 2017.

[20] Z. Chen, L. Tian, and C. Lin, "Trust model of wireless sensor networks and its application in data fusion," *Sensors*, vol. 17, no. 4, p. 703, Mar. 2017.

[21] G. Spanos, K. M. Giannoutakis, K. Votis, and D. Tzovaras, "Combining statistical and machine learning techniques in IoT anomaly detection for smart Homes," in *Proc. IEEE 24th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2019, pp. 1–6.

[22] L. Fang and S. Dobson, "Unifying sensor fault detection with energy conservation," in *Proc. Int. Workshop Self-Organizing Syst.*, 2013, pp. 176–181.

[23] F. Cauteruccio, G. Fortino, A. Guerrieri, and G. Terracina, "Discovery of hidden correlations between heterogeneous wireless sensor data streams," in *Proc. Int. Conf. Internet Distrib. Comput. Syst.* Cham, Switzerland: Springer, 2014, pp. 383–395.

[24] J. Choi, H. Jeoung, J. Kim, Y. Ko, W. Jung, H. Kim, and J. Kim, "Detecting and identifying faulty IoT devices in smart home with context extraction," in *Proc. 48th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw. (DSN)*, Jun. 2018, pp. 610–621.

[25] K. Kapitanova, E. Hoque, J. A. Stankovic, K. Whitehouse, and S. H. Son, "Being SMART about failures: Assessing repairs in SMART Homes," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 51–60.

[26] A. K. Sikder, H. Aksu, and A. S. Uluagac, "6thSense: A context-aware sensor-based attack detector for smart devices," in *Proc. USENIX Secur. Symp.*, 2017, pp. 397–414.

[27] N. Berjab, H. H. Le, C.-M. Yu, S.-Y. Kuo, and H. Yokota, "Hierarchical abnormal-node detection using fuzzy logic for ECA rule-based wireless sensor networks," in *Proc. IEEE 23rd Pacific Rim Int. Symp. Dependable Comput. (PRDC)*, Dec. 2018, pp. 289–298.

[28] K. Kapitanova, S. H. Son, and K. D. Kang, "Using fuzzy logic for robust event detection in wireless sensor networks," *Ad Hoc Netw.*, vol. 10, no. 4, pp. 709–722, 2012.

[29] D. Goldin and P. C. Kanellakis, "On similarity queries for time-series data: Constraint specification and implementation," in *Proc. Constraint Program.*, 1995, pp. 137–153.

[30] J. Paparrizos and L. Gravano, "K-shape: Efficient and accurate clustering of time series," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1855–1870.

[31] J. Miller, "Reaction time analysis with outlier exclusion: Bias varies with sample size," *Quart. J. Exp. Psychol. A*, vol. 43, no. 4, pp. 907–912, 1991.

[32] Y. Sakurai, S. Papadimitriou, and C. Faloutsos, "BRAID: Stream mining through group lag correlations," in *Proc. ACM SIGMOD Int. Conf. Manage. data (SIGMOD)*, 2005, pp. 599–610.

[33] *Intel Lab Data*. Accessed: Mar. 22, 2021. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

NESRINE BERJAB was born in Tunis, Tunisia. She received the B.S. degree in applied network infrastructure and system administration from the National Engineering School of Carthage, Tunis, in 2012, the Dipl.-Ing. degree in software engineering from The University of Tunis El Manar, Tunis, in 2015, and the M.E. degree in computer science from Tokyo Institute of Technology, Tokyo, Japan, in 2019, where she is currently pursuing the Ph.D. degree. Her current research interests include data engineering and secure and dependable IoT systems.

HIEU HANH LE received the B.S., M.E., and Dr.Eng. degrees from Tokyo Institute of Technology, in 2008, 2010, and 2015, respectively. He was a Researcher at Yokohama Research Laboratory, Hitachi Ltd. His research interests include data engineering, information storage systems, privacy, information retrieval, and network engineering. He is a member of ACM SIGMOD, IEEE CS, IPSJ, IEICE, and DBSJ.

HARUO YOKOTA (Senior Member, IEEE) received the B.E., M.E., and Dr.Eng. degrees from Tokyo Institute of Technology, in 1980, 1982, and 1991, respectively. He joined Fujitsu Ltd., in 1982, and was a Researcher at ICOT for the 5th Generation Computer Project, from 1982 to 1986, and at Fujitsu Laboratories Ltd., from 1986 to 1992. From 1992 to 1998, he was an Associate Professor at Japan Advanced Institute of Science and Technology (JAIST). He moved to Tokyo Institute of Technology, in 1998, where he is currently a Full Professor and the Dean of the School of Computing. His research interests include the general research areas of data engineering, information storage systems, and dependable computing. He is also a Board Member of DBSJ, a fellow of IEICE and IPSJ, and a member of IFIP-WG10.4, JSAI, JAMI, ACM, and ACM-SIGMOD. He has been the Chair of ACM SIGMOD Japan Chapter, the Vice President of DBSJ, a Trustee Board Member of IPSJ, the Editor-in-Chief of *Journal of Information Processing*, and an Associate Editor of *The VLDB Journal*.