Fuzzy logic and ECA rules-based misbehaving-node detection for cluster-based heterogeneous IoT systems

Nesrine BERJAB†, Hieu Hanh LE†, Chia-Mu YU††, Sy-Yen KUO†††, and Haruo YOKOTA†

† Tokyo Institute of Technology, 2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
†† National Chung Hsing University, 145 Xingda Road, South District, Taichung City 40227, Taiwan
††† National Taiwan University, No. 1, Section 4, Roosevelt Road, Taipei 10617, Taiwan
E-mail: †{berjab,hanhlh}@de.cs.titech.ac.jp, ††chiamuyu@gmail.com, †††sykuo@ntu.edu.tw, ††††yokota@cs.titech.ac.jp

Abstract The IoT devices have been the subject of much interest. Nevertheless, these devices are resource constrained and susceptible to false-data-injection attacks and failures, leading to unreliable and inaccurate sensor readings. In this paper, we propose a hierarchical framework for detecting misbehaving nodes in WSNs. It uses fuzzy logic in event-condition-action (ECA) rule-based WSNs to detect malicious nodes, while also considering failed nodes. The spatiotemporal semantics of heterogeneous sensor readings are considered in the decision process to distinguish malicious data from other anomalies. Our experiments using real-world dataset demonstrate that our approach can provide high detection accuracy with low false-alarm rates.

Key words Internet of things, wireless sensor network, security, misbehaving node detection, fuzzy logic, sensor correlation, ECA rules

1. Introduction

The Internet of Things (IoT) can be described as a dynamic and distributed networked system that uses wireless connectivity and is composed of a wide range of uniquely identifiable embedded computer-like devices. One of the essential elements of the IoT paradigm is the wireless sensor network (WSN). WSNs are composed of smart-sensor nodes that monitor their environmental conditions, report sensor data, and perform appropriate actions in response to the surrounding circumstances.

However, these sensor nodes suffer from resource constraints such as processing power, memory, and energy supply. Moreover, because of the absence of appropriate high-level abstractions to simplify the programming of WSNs, application development remains challenging. In the IoT domain, "If-This-Then-That" is an example of an abstraction. It is a simple rule that triggers an action if a particular event occurs. For example, "If the room temperature increases, then regulate the air conditioner to cool the room." However, the interaction between the two devices involved has a security issue. Decisions are taken by considering only the output of the devices without observing whether their current operational state is normal or on a state of misbehaving. Such incomplete specifications will lead to inaccurate and unreliable sensor readings that may lead to incorrect decisions and even to real-world damage. Indeed, sensor nodes are often exposed to open or hostile environments. This makes it easy for attackers to compromise some of the sensor nodes and manipulate the integrity of the sensed data, e.g., by injecting fake packets. As far as we know, the area of false-data-injection attacks (FDIAs) detection for IoT is yet to receive the attention it deserves. Most previous intrusion detection methods proposed for IoT, particularly for WSNs, focus only on specific types of network attack. Few approaches have included an efficient, adaptive WSN intrusion-detection application that considers methods for programming the sensor nodes.

To answer these challenges and to guarantee reliable monitoring in WSNs, we propose a new hierarchical framework based on fuzzy logic for detecting misbehaving nodes in event-condition-action (ECA) rule-based WSNs. Our contribution is to provide an integrated solution for programming the sensor nodes and distributedly detecting misbehaving nodes in hierarchical heterogeneous WSNs. By controlling the sensor nodes according to a set of ECA rules, we can better express network behaviors and detect malicious nodes while considering failed nodes. Identification of failed nodes is the first step in countering the threats against WSN reliability. In this paper, we consider that when a node fails, it stops sensing the environment and sending report messages. Therefore, based on a preliminary analysis of potential fail-

ure in sensor nodes, the values for newly collected sensed data are further analyzed to increase the attack detection rate and reduce the false positive rate. To achieve this, we use a fuzzy logic module [1] to treat the ambiguity in the decision-making process. It arrives at a conclusion based on spatiotemporal (ST) and multivariate attribute (MVA) sensor correlations to distinguish malicious sensor data from other anomalies. This proposed fuzzy logic-based detection is our first contribution in this paper. It can better detect FDIAs attacks because the rule base contains a better set of rules.

Because of the various limitations of synthetic datasets, in this paper, we use a specialized real-world dataset for WSNs that was collected to evaluate the detection efficiency of our approach under realistic conditions. The evaluation involves real WSN data rather than simulated data, and this is our second contribution in the paper. Following our experiments with the proposed framework, we stress the significance of considering fuzzy logic and the sensor correlations to achieve a higher detection accuracy, which has been neglected in previous studies. Our experiments demonstrate that our approach can provide high detection accuracy with low false-alarm rates. We also show that our approach performs well when compared to two well-known classification algorithms.

The remainder of this paper is organized as follows. We provide an overview of existing approaches in Section 2. Section 3 describes some preliminary ideas before introducing our approach. Our proposed scheme is presented in Section 4. Our experiments and results are presented in Section 5, together with a performance evaluation. Finally, Section 6 concludes the paper and describes possible future work.

This paper is a revised version of a paper entitled Hierarchical Abnormal-node Detection using Fuzzy Logic for ECA rule-based Wireless Sensor Networks by the same authors. The earlier version was presented at the IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC) [2].

2. Related Work

Failures and network detections for IoT systems have been explored a lot over decades. However, as far as we know, the area of FDIAs for WSN is yet to receive much attention. Moreover, they usually rely on using precise values to specify anomalies thresholds and ignore consideration of the dependencies that exist between sensor data. Indeed most of the exisitng works uses either ST correlations, and few have considered the MVA correlations. In [3,4], the authors use spatial correlation to detect events within WSNs, without considering the possibility of the presence of malicious or failed nodes. Therefore, the false-alarm rate is high because malicious or abnormal readings are also considered as events. In [5], the authors propose distributed anomaly detection by assuming an environment monitored by homogenous sensors. A monitored area should involve not only a set of homogenous sensors but also some heterogeneous sensors. By capturing the MVA sensor correlations, we can confirm whether the attributes collectively imply an anomaly.

Moreover, the detection architectue plays an important role the the accuracy of the analysis. In [5], the authors adopt a traditional, centralized approach to detection. It needs to collect a significant volume of sensed data. The larger volume of the collected data generated by centralized detection, results in lack of scalability and usually introduces a delay between the moment the data is collected and when the data intrusion system can have access to them. In [4], a rule-based distributed fuzzy inference system for WSNs is proposed that uses the spatial correlation (ST) from the observed sensor readings, to identify the occurrence of events. Their experimental results showed that using a fuzzy logic-based distributed detection improves the accuracy of the event detection. However, each node was required to exchange and process a significant amount of data from neighboring nodes. Therefore, communication and computation overheads are introduced, which will decrease the usage time of the sensor nodes' batteries.

Nevertheless, none of these approaches considered methods for developing adaptive WSN applications. However, a few proposals for efficient WSN applications consider methods for programming the sensor nodes. In [7], the author introduces increased flexibility in programming WSNs, but the proposed solution involves significant overheads. In [8], an ECA rule-based model is used to describe business rules for sensor nodes, which are designed to express service-oriented business logic in a compact form.

Concretely, we propose a new integrated approach for programming the sensor nodes and distributedly detecting the misbehaving-nodes in hierarchical heterogeneous WSNs. The detection process is controlled by ECA rules, and the spatiotemporal semantics correlations of heterogeneous sensor readings are considered in the decision analysis.

3. Preliminaries

In this section, we describe the characteristics of our proposed framework and discuss some assumptions about the monitoring environments considered in this paper.

3.1. System model

An environmental monitoring application in a WSN is defined as an application that monitors the real world and is-

sues a report whenever an event of interest occurs. The application for environment monitoring considered here has several heterogeneous sensor nodes deployed wirelessly within the monitored areas. This paper addresses the network-

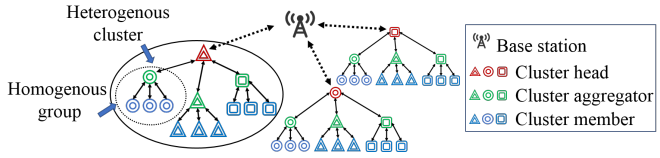

Figure 1: Hierarchical distributed WSN based on two-level clustering

scalability issue by adopting a hierarchical, distributed WSN topology based on two-level clustering. Figure 1 depicts the topology of this hierarchical network. To implement such a thorough monitoring system, n sensor nodes (S_1, S_2,..., S_n) called cluster members (CMs), are grouped into a homogenous group according to their type. Each homogeneous group is controlled by a cluster aggregator (CA), who is responsible for all communications between the CM nodes and the upper level of the hierarchy. These homogenous groups are separated into clusters, each covering a different area. Each cluster should include one cluster head (CH), who is responsible for all communications between the CA nodes and the base station (BS). All the sensor nodes have two primary functionalities: i.e., environmental monitoring and data aggregation. Each CM monitors the environment, collects newly sensed data and reported to the higher level. Each CA and CH aggregates the received reports from the lower level (CM and CH, respectively) and then send a combined report to a higher-level node (CH or BS, respectively). By adopting this topology, the network scalability for large-scale WSNs can be achieved. Although several complex and innovative clustering techniques have been proposed for WSNs, we consider a very simple clustering technique for environmental monitoring in WSNs. The clustering formation is based on a defined distance threshold between the sensor nodes.

In our WSN, the sensor nodes are controlled according to a set of ECA rules. ECA rules are widely used for controlling an environment and act to control the system configuration. ECA rules comprise a set, whereby each rule in the set reacts to a detected *event* by evaluating a *condition* and executing an *action* whenever the event happens and the condition is verified [9]. In our system, the set of ECA rules describe the behavior of the sensor nodes and perform the required actions only when the relevant events happen and their conditions are verified. The ECA rules are independent of other rules in the system (fuzzy rules). The structure of each rule is ON Event IF Condition THEN Action.

3.2. Permanent Failures and FDIAs model

Our objective is to detect misbehaving sensor nodes. In this paper, we mainly focus on detecting permanent failures. A permanent failures is when there is a hardware defect, and the sensor node completely shuts down. We also cover the problem of malicious attack detection. Precisely, the issue of FDIAs detection. FDIAs is a significant threat and crucial for the case of systems where the actuator takes decisions based on the collected sensor data [10]. Furthermore, it is known for its difficulty to be detected by the conventional IDS. If the attacker can recognize the standard conditions of the monitored environment or some system parameters, he can easily inject false data into the regular sensor readings without being detected by the IDS. In this paper, we consider two types of realistic attack goal such as random FDIAs, in which the attacker aims to compromise the sensor readings data randomly, and specefic FDIAs, in which the attacker injects a specific value within the pre-defined range.

3.3. Assumptions

Our research is based on the following assumptions.

1) Every sensed environment is characterized by its own environmental conditions.

2) All clusters must be composed of both homogeneous and heterogeneous sensor nodes, to maintain high event-detection accuracy.

3) N-modular redundancy is used to achieve a dependable and fault-tolerant WSN. The considered WSN must satisfy a good distribution of the clusters, where at least three sensor nodes for each type must be deployed within one cluster (i.e., triple modular redundancy (TMR) is a particular case of N-modular redundancy).

4) The attacker knows the physical conditions of the monitored environment.

5) While some sensor nodes may be compromised, we assume that most of the sensor nodes will remain trustworthy. Moreover, we assume that any sensor node might become compromised and behave maliciously, except for the BS, CHs and CAs.

4. Proposal

The proposed approach needs to consider changes in the environmental conditions, perform appropriate actions to identify malicious and failed nodes, and report such nodes to the BS. In this paper, we propose a new framework for detecting malicious nodes in a heterogeneous WSN while considering the failed nodes. The proposed approach adopts a hierarchical detection framework based on fuzzy logic for ECA rule-based WSN. By defining a set of ECA rules, we can describe the behavior of the sensor nodes and identify misbe-

having nodes. Figure 2 shows an example of the ECA rules used for our WSN. If the sensed data value is outside a predefined range, this node is reported as an misbehaving node. Alternatively, if there is no data reported from a sensor node, then this node is reported to the BS as having failed. However, it is infeasible for the sensor node to decide that any abnormal data detected originates from attacks using only simple rules and the raw sensor node data. Therefore, based on the preliminary failure analysis, the value of sensed data should be further analyzed to increase the malicious-node detection rate and reduce the false-positive rate. To achieve this, we add fuzzy logic to the detection module to treat the ambiguities in the decision-making process. By considering the MVA and ST sensor correlations, the fuzzy rule base will contain a better set of rules to derive a conclusion about whether a sensor node is malicious.

```
ON new_ temperature IF (temperature_get <> NULL) DO local_detction
ON crisp_localDetection IF true DO report_decision_to_CAnode
ON received_localreports IF (localDecisions_get = NULL) DO report_failure_to_BS
ON received_localreports IF true DO group_detection
ON crisp_groupDetection IF true DO report_decision_to_CHnode
ON received_globalreports IF (globalDecisions_get = NULL) DO report_failure_to_BS
ON received_groupReports IF true DO cluster_detection
ON crisp_clusterDetection IF true DO report_attack_to_BS
```

Figure 2: ECA rule-based WSN description example

An overview of our proposed approach is depicted in Figure 3, which shows the various sensor-node modules and the flow chart for processing misbehaving nodes according to the role of the CH and CA sensor nodes. Each CM monitors the environment and collects newly sensed data. Each CA and CH aggregates the received reports from the lower level (CM and CH, respectively) and then sends a combined report to a higher-level node (CH or BS, respectively).

Each sensor node has a misbehaving-node detection module. This module has two submodules. The first is a failure-detection module that checks whether there are permanent failed nodes. Sensor nodes may be prone to many types of hardware failure induced by a power surge, weak batteries, loss of signal, or corruption of the external memory. As a result, the sensor nodes will be unable either to sense data or report sensed data to higher-level sensor nodes. If a node stops sending reports, the higher-level node concludes that the sensor node has incurred a permanent failure and reacts by reporting it to the BS. The BS then may issues instructions about reorganizing the grouping or clustering arrangements.

The second misbehaving-node detection submodule is a fuzzy-based detection module that checks for malicious nodes. The detection process involves three stages of detection: i.e., *local detection*, *group detection*, and *cluster detection*. As illustrated in Figure 3, each CM monitors the environment and collects newly sensed data. The CM then analyzes this data by executing the fuzzy-based local-detection process. This local decision about maliciousness will be reported to the CA. After the CA receives all the reports from its CMs (homogenous sensor nodes), the CA executes the fuzzy-based group-detection process, to create a more accurate decision. By considering the CMs' local decisions in its group, the CA generates the group decision and reports it to the CH. After the CH receives all the reports from its CAs (heterogeneous sensor nodes), the CH executes the fuzzy-based cluster-detection process. This process' final decision will be reported to the BS. The three fuzzy detection modules are described in detail in the following subsections.

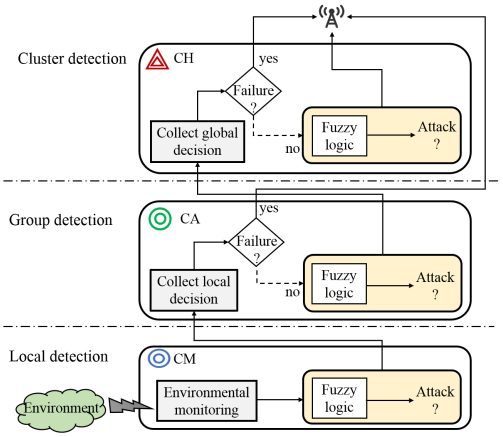

Figure 3: Hierarchical detection modules according to the role of each sensor node

4.1. Local detection by the CMs

Each CM senses environmental events and executes the local detection process to check whether the newly collected data is subject to attack.

4.1.1. Temporal average similarity

The local detection module considers temporal semantic correlations to derive a crisp local decision. Every CM maintains a short-term history of the collected sensed data. This aggregation of data is used to construct a sliding time window containing the most-recent sensed data in the sensor-node stream. In the literature of stream processing, sliding time windows are a familiar concept [11]. In this paper, we use the sliding time window to profile the behavior of the sensor node readings over time. The sensed data will be time correlated and the variation range will usually be small in the short term. The sensed data contains an k-second timestamp, which indicates the time at which the sensor node reported the reading. The sensor node time-series samples are grouped into $(l+1)$-second frames to compose the sliding time-window model, where $l \in \{0, p\}$. As time passes, the

window slides in one-frame increments over the sensor node time series. Each frame groups the raw sensor node-data samples according to a certain number of epochs e_k, where $k \in \{1, o\}$. After setting the sliding time window, we apply a summarization function to extract the relevant information about sensor-node temporal similarity. For example, we consider a sliding window composed of three frames. Each frame is composed of three epochs. f_0 is the frame containing the current epoch. f_1 is the frame for the o previous epochs. Let $l+1$ be the size of the sliding time window and k be the number of epochs within each frame. For each frame f_l within the window, we calculate the temporal similarity between the frame f_l and the current frame f_0. The temporal similarity is given by equation (1).

$$q(f_{-l}, f_0) = \frac{1}{(1 + \sqrt{\sum_{k=1}^{o} S_i(e_k)_{f\_l} - S_i(e_k)_{f_0})^2}))} \quad (1)$$

The average similarity between the current frame data and the data in the window is then calculated. As indicated in equation (2), the average similarity is calculated by adding a weighted summation to the calculation. The closer the frame is to the current time frame, the more it is correlated. Moreover, the smaller the average similarity, the more the frame at the current time deviates from the historical sensor node data.

$$Q(f_0) = \frac{\sum_{l=1}^{p} w_l q(f_{-l}, f_0)}{l+1} \quad (2)$$

where the weight: $w_l = \frac{1}{(e_o)_{f_0} - (e_o)_{f_{-l}}}$

4.1.2. Fuzzification and Defuzzification

After the CM finishes the calculation of the average temporal similarity, it conducts the fuzzy local-detection process. Together with the temporal average-similarity value obtained, both the current raw sensed value $S_i(t)$ and its actual time of collection are fuzzified through predefined membership functions (MFs). The monitored environment can differ for each time-of-day segment. As a result, the input–output response will also differ depending on the time of day. For example, the light intensity and temperature during the day tend to be higher than at night. Figure 4 illustrates details of the design of the adopted scheme for local detection of FDIAs. In our system, we consider an environment

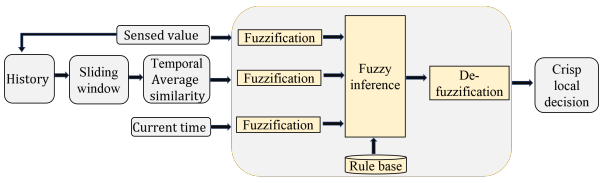

Figure 4: Local detection scheme for CMs

Table 1: Rule base for temperature sensor nodes

Rule ID	Temperature	TAS	CT	Malicious
1	VL	S	Mo	H
2	VL	B	A	H
3	MH	B	A	L
4	MH	S	N	H
5	LM	S	N	M
6	LM	S	A	H
⋮	⋮	⋮	⋮	⋮
32	MH	B	M	L

monitored by sensor nodes for temperature, humidity, light, and smoke density. For each type of sensor node, the local detection module takes three linguistic variables as its input. The sensed-value input will be one of temperature, humidity, light, or smoke. In the fuzzification process, the three crisp values are converted into degrees of membership by applying the corresponding MF. The MF for the temperature variable has four semantic values: i.e., very low *VL*, low-to-medium *LM*, medium-to-high *MH*, and very high *VH*. The MFs for the humidity, light, and smoke variables have three semantic values: i.e., low *L*, medium *M*, and high *H*. The MF for average temporal similarity *TAS* has 2 semantic values: i.e., small *S* and big *B*. Finally, the MF for the current time *CT* has 4 semantic values: i.e., night *N*, morning *Mo*, afternoon *A*, and evening *E*. The confidence about malicious detection is defined as the output. The MF for the fuzzy output variable is defined in terms of three levels: i.e., low *L*, medium *M*, or high *H*.

After being fuzzified, the fuzzy inputs are then fed into the fuzzy inference process. The fuzzy rule base manages the inference to yield a fuzzy output. A fuzzy rule base comprises a set of rules designed to decide the probability of the node being compromised. By considering the ST and MVA sensor correlations, we use heuristics to build the rule base for our malicious-detection experiments. However, if more-complex attacks are to be detected, domain experts or machine learning techniques could be used to define the rule base.

The form of these rules is "IF premise, THEN consequent," where the premise is the fuzzy input variables connected by logical functions and the consequent is the fuzzy output variable. An example might be "IF *temperature* is *H* AND *TAS* is *S* AND *CT* is *N*, THEN *Malicious* is *H*." The full rule base for the four types of sensor node involves 104 rules. The rule base of temperature sensor nodes in WSNs is shown in Table I. This rule base contains only the rules involving linguistic variables based on the sensed values from temperature sensor nodes. The rule base for the other sensor node can be constructed similarly. The temperature sensor node's rule

base involves only 32 rules and the light, smoke-density, and humidity sensor nodes' rule bases each involve only their own subset of 24 rules. Finally, the defuzzifier converts the output fuzzy variable back to a crisp value, which is used to make a local decision and send a report message to the CH for the further analysis that eventually leads to a groupl decision.

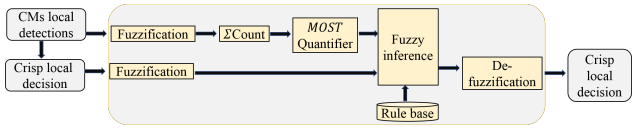

Figure 5: Group detection scheme for the CA

4.2. Group detection by the CA

The group detection module operates at the CA level by considering ST correlations. In this detection module, a more accurate decision is made by including the local decisions of multiple homogenous CMs located in the same group within the same cluster. After receiving a local decision message from a CM, the CA stores the crisp decision value. After collecting all the CMs' local decisions, the CH executes the cluster-detection process for each CM node to give a group decision about the node's maliciousness. Figure 5 illustrates the details of the design of the adopted scheme to detect malicious nodes within a group. The group detection module uses two inputs: i.e., every individual CM's crisp local decision and all the CMs' local decisions. The fuzzifier converts the crisp values into degrees of membership by applying the corresponding MF. After being fuzzified, a sigma-count factor [12] is used as a measure of fuzzy cardinality to quantify the CMs' local decisions:

$$\sum Count(F) = \sum \mu F(S_i) \tag{3}$$

Here, F is a fuzzy set characterized by an MF $\mu F(S_i)$, which gives the degree of similarity for S, and $S_i = (S_1, S_2, \ldots, S_n)$ is the set of CMs. Finally, F is the property of interest related to the sensor-node's local decision, e.g. "Misbehavior level is high." A fuzzy majority quantifier is then used to obtain a fuzzified indication of the consensual CMs' local decisions. For a more accurate decision, we use the *Most* quantifier to characterize the fuzzy majority of the CMs' local decisions [13]:

$$u_{most}\left(\frac{\sum Count(F)}{|S_i|}\right) = u_{most}\left(\frac{\sum_i \mu F(S_i)}{n}\right)$$
$$\text{where } u_{most}(x) = \begin{cases} 0 \text{ if } x \leqq 0.3; \\ 2x - 0.6 \text{ if } 0.3 < x < 0.8 \\ 1 \text{ if } x \geqq 0.8 \end{cases} \tag{4}$$

Next, the fuzzified inputs and the quantified CMs' local decisions are fed into the fuzzy inference process. The fuzzy rule base comprises a set of rules designed to decide about the maliciousness of the CM. An example of the format of the rule is " IF *Malicious* is *H* AND *Most(CMsDecision)* is *L* THEN *Malicious* is *H*." Fuzzy inference combines the rules to obtain an aggregated fuzzy output. Finally, the defuzzifier converts the fuzzy output variable back to a crisp value that will be used to make a group decision and reported to the CH.

4.3. Cluster detection by the CH

Cluster identification is processed in the CH level by considering the ST and MVA sensor correlations. In this detection module, a more accurate decision is made by including the group decisions of multiple heterogenous CAs located in the same cluster. After receiving a group-decision message from a CA, the CH stores the crisp decision value. After collecting all the CA group decisions, the CH performs the fuzzy inference for each sensor node to give the cluster decision about the node's maliciousness. The detection mechanism is similar to that for group detection. However, compared to the group decision, the cluster decision considers the observations from heterogeneous-sensor nodes in addition to only homogenous-sensor nodes. The CH's fuzzy rule base comprises a set of rules designed to decide about the CM's maliciousness. An example of the rule might be "IF *Malicious* is *L* AND *Most(CAsDecision)* is *L* THEN *Malicious* is *L*. If malicious nodes are detected, the CH sends a report message to the BS.

5. Experimental Results and Analysis

There are various limitations to using synthetic datasets and none of the previously proposed misbehaving-node-detection approaches have yet been developed as deployable systems. In this paper, we evaluate the detection efficiency of our approach under realistic conditions. Sixteen sensor nodes were deployed in our laboratory using the Raspberry Pi 2 Model B microcontroller platform. Each sensor node is equipped with one temperature sensor module, one digital light-intensity sensor, one smoke-density sensor, and one humidity sensor, which gives a total of 64 sensors. The sensor nodes were divided into three clusters separated from each other and with different environmental conditions. One cluster comprised five sensor nodes located in our laboratory room, the second was composed of five sensor nodes located in a server room, and the third was composed of six sensor nodes located in a kitchen corner.

5.1. WSN dataset creation and description

To evaluate the detection efficiency of our approach, we collected a real-world dataset over a period of one month. New sensor readings were collected every minute, giving a

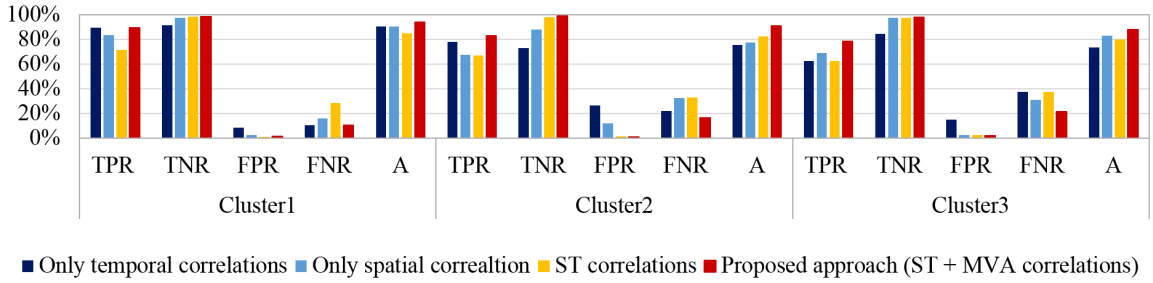

Figure 6: Comparison of the detection results with a single correlation or a combination of correlations in each cluster

total of 2.787.776 sensor readings. The collected dataset contains 93% real normal readings, 4.34% real failures and 2.66% artificial injected false sensor readings.

This dataset was obtained after we preprocessed the collected sensor node data. There were two significant challenges in the preprocessing. Because of individual failures, some sensor nodes had missing data for epochs in the dataset. This meant that data entries would not always be consecutive. Therefore, the value "NULL" was used to substitute for each missing sensed value in an epoch. Moreover, to investigate the performance of our approach, malicious readings were injected to the collected dataset according to the attack patterns described in section 3.2. We altered the readings for a certain number of sensor nodes by a certain amount in specific time slots.

5.2. Evaluation of results

Our proposed approach perfectly detected all the failed nodes in our deployed WSN. However, to evaluate our approach in terms of malicious-node detection, five performance metrics are used in this paper: i.e., *true positive rate* (TPR), *true negative rate* (TNR), *false positive rate* (FPR), *false negative rate* (FNR), and *accuracy* (A). Figure 6 summarizes the results of the metrics for each cluster. Even though the environmental conditions for each cluster were different, our proposed approach achieves a high accuracy with a small FPR for the task of analyzing the sensor readings to determine whether the sensor nodes were behaving normally or had been exposed to attacks. The rate of correctly classified readings (TPR and TNR) was higher than the number of incorrectly classified readings (FPR and FNR).

In addition, to stress the significance of considering sensor correlations and the hierarchical architecture in achieving better detection accuracy, we conducted other experiments whereby only a single correlation or a combination of two-correlations are considered. The experimental results show that our approach produces better detection results compared to the other cases. The first case is where each CM performs its local detection and then send its decision directly to the BS. That is, only temporal correlation was considered. For the only spatial correlation case, only the group detection is performed by the CA. And as for the ST correlation, only the local and group detections are performed. The detection results show that our hierarchical approach (ST +MVA correlations) demonstrates higher accuracy and lower false-alarm-rate than in the other cases.

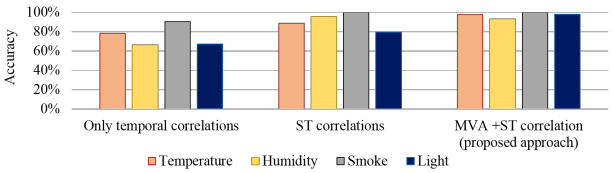

Figure 7: Accuracy results in cluster 2 where a single correlation or a combination of two-correlations are considered

Figure 7 depicts details of accuracy results for each sensor node type in cluster 2. As shown, not all sensor nodes types show high accuracy, especially when only temporal or spatial correlation is considered. For example, the average accuracy of the light sensor nodes in Cluster 2 is low, because the light intensity in the server room is usually very low but, according to the defined fuzzy rules for the morning period, we should expect high light intensity. As a result, the false negative rate was high. However, the results show that we can decrease the false-alarm rate and improve the detection accuracy by including more than one correlation in the decision process. In our approach, when most of the sensor nodes show abnormal readings, it implies that they are not under malicious attack but there may be an event in the monitored environment. Therefore, the detection accuracy was higher when the ST and MVA sensor correlations are included. To

Table 2: Incorrect classification introduced by Naive Bayes, J48 and our proposed approach

	Naïve Bayes	**J48**	**Proposed approach**
Cluster 1	18.56%	1.21%	1.43%
Cluster 2	13.72%	0.67%	1.65%
Cluster 3	50.84 %	1.39%	2.45%

further understand the behavior of our approach, we compared it to two well-known classification algorithms: a naïve Bayes classifier and a j48 decision tree. Fuzzy logic is more appropriate for describing WSN behavior than these two algorithms. This is unlike Bayes classifiers and decision trees, where values are considered to be discrete, fuzzy logic works with continuous values, which are precisely what the sensor readings are. We used the WEKA data-mining tool to run this experiment. The attribute values supplied to the classification algorithms were the same raw input data used for the fuzzy module in our approach (i.e., temperature, light intensity, humidity, smoke density, and their average temporal similarity.) In the evaluation, we performed a 10-fold cross validation for both classification algorithms.

Table 2 shows the number of incorrectly classified instances for the three clusters. Both algorithms wrongly classified a certain number of instances. Note that naïve Bayes introduced the most incorrect classifications. Our proposed approach had a lower percentage of incorrectly classified instances than naïve Bayes, but it was higher than that for the J48 decision tree. However, compared to these other two classification algorithms, fuzzy logic is better suited to WSNs, not only because it works with continuous readings of the sensor nodes, but also because it only needs to specify MFs and rules, which is more straightforward and computationally more efficient than having to build complex probability models. Besides, naïve Bayes and J48 not only needs time to train their model, but also they need to split the dataset into testing and training data. While our approach does not need any training data to perform the detection process.

6. Conclusion

In this paper, we propose a novel hierarchical approach to detect misbehaving nodes in WSNs. The proposed approach uses fuzzy logic for ECA rule-based WSNs to detect malicious nodes while considering failed nodes. The ST semantics of heterogeneous sensor readings are considered in the decision process to distinguish malicious data from other anomalies. Our experiments on real-world sensor data demonstrate that our approach can provide high detection accuracy with low false-alarm rates. The experiments also support the hypothesis that including ST and MVA sensor correlations in the decision process further improves the malicious-nodes detection accuracy. In addition, when compared to two well-known classification algorithms, our proposed approach performed well. In our future work, we plan to implement a mechanism that derives the fuzzy rule set automatically, perhaps via machine learning techniques, to improve the detection accuracy and decrease the false-alarm rate. In addition, we aim to further investigate the detection accuracy by considering the setting of the thresholds such as the distance-based clustering.

Acknowledgment

This research was supported by JST as part of the Japan-Taiwan Collaborative Research Program and by MOST as part of 105-2923-E-002-014-MY3.

References

[1] L.A. Zadeh, "Fuzzy Sets," in Information and Control, Volume 8, Issue 3, pp. 338–353, (1965)

[2] N. Berjab, C.M. Yu, S.Y. Kuo, and H. Yokota, "Hierarchical Abnormal-node Detection using Fuzzy Logic for ECA rule-based Wireless Sensor Networks," in Proceedings of the IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC), pp. 289–298, (2018)

[3] K. Kapitanova, S.H. Son, and K.D. Kang, "Using Fuzzy Logic for Robust Event Detection in Wireless Sensor Networks," in Ad Hoc Networks, Volume 10, Issue 4, pp. 709–722, (2012)

[4] M. Marin-Perianu and P. Havinga, "D-FLER: A Distributed Fuzzy Logic Engine for Rule-Based Wireless Sensor Networks," in Proceedings of the 4th International Conference on Ubiquitous Computing Systems (UCS), pp. 86–101, (2007)

[5] S. Rajasegarar, C. Leckie, M. Palaniswami, and J.C. Bezdek, "Distributed Anomaly Detection in Wireless Sensor Networks," in Proc. IEEE International Conference on Communications, ICCS, (2006)

[6] B. Sheng, Q. Li, W. Mao, and W. Jin, "Outlier Detection in Sensor Networks," in MobiHoc '07, Proceedings of the 8th ACM International Symposium on Mobile ad hoc Networking and Computing, pp. 219–228, (2007)

[7] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks," in International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 85–95, (2002)

[8] M. Marin-Perianu, T.J. Hofmeijer, and P.J.M. Havinga, "Implementing Business Rules on Sensor Nodes," in 11th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), pp. 292–299, (2006)

[9] A. Paschke, "ECA-RuleML: An Approach Combining ECA Rules with Temporal Interval-Based KR Event/Action Logics and Transactional Update Logics," in Computer Research Repository, abs/cs/061, (2006)

[10] N. Berjab, C.M. Yu, S.Y. Kuo, and H. Yokota, "Impact Analysis for Dos and Integrity Attacks on IoT Systems," in Proceedings of the 7th International Conference on Information Systems and Technologies, pp. 1–8, (2017)

[11] 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 16th IEEE Intl. Conf. on Dependable, Autonomic, and Secure Computing, pp. 568-575, (2018)

[12] J. Kacprzyk, "Group Decision Making with a Fuzzy Linguistic Majority," in Fuzzy Sets and Systems, 18(2), pp. 105–118, (1986)

[13] L.A. Zadeh, "A Computational Approach to Fuzzy Quantifiers in Natural Languages," in Computers and Mathematics 9, pp. 149–184, (1983)