Modeling and Simulation of Abnormal Behavior Detection Through History Trajectory Monitoring in Wireless Sensor Networks

JUNG SUB AHN\textsuperscript{1} AND TAE HO CHO\textsuperscript{2}

\textsuperscript{1}Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, South Korea
\textsuperscript{2}College of Computing and Informatics, Sungkyunkwan University, Suwon 16419, South Korea
Corresponding author: Tae Ho Cho (thcho@skku.edu)

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2021R1A2C2005480).

ABSTRACT Data security is becoming increasingly important with the growing popularity of smart management systems combining wireless sensor networks (WSNs) and intelligent systems in industrial, agricultural, and construction fields. Traditional security methods for WSN have focused on data integrity and the identification of outliers in sensor data and are therefore vulnerable to attacks such as false positive attacks (FPAs) and false negative attack (FNAs) using massively compromised nodes. These attacks significantly compromise the sensor nodes because the correlation between node verification behaviors is not considered in the communication process until the intercepted sensor data are used as input into the system. This study introduces an FPA and FNA detection method using a spatiotemporal historical data-based knowledge base. The main contribution of the present study is the recognition of abnormal correlations using behavior monitoring through the discrete event system specification model. By recognizing abnormal correlations, the proposed method prevents the inflow of false data generated as a result of widespread damage. Furthermore, a new strategy is proposed to maximize the lifetime of the network by blocking any compromised nodes. The proposed spatiotemporal data-based detection approach is applicable to a wide variety of fields owing to its use of a shared security model. The proposed method was shown to reduce the energy consumption by 46.737% and 41.927% in comparison to statistical en-route filtering and cluster-based false-data filtering, respectively.

INDEX TERMS Discrete event system specification (DEVS), false negative attack detection, false positive attack detection, history trajectory monitoring, network security, wireless sensor network.

I. INTRODUCTION

Wireless sensor networks (WSNs) collect sensor data from various nodes and conduct monitoring in various domains pertaining to the environment, healthcare, agriculture, and military tracking [1], [2], [3], [4], [5], [6], [7], [8]. WSN-based smart management systems such as smart furnaces [4], smart semiconductors [9], logistics [10], and robotics [7], [11] collect information [12], [13], [14] and make decisions using the collected data. A WSN consists of base stations for data collection and numerous sensor nodes for sensing. To reduce costs, most sensor nodes expend low energy and have limited transmission bandwidth and memory capacity [15], [16]. In addition, a sensor node using the TCP protocol requires a large number of computational resources; therefore, in the best model, data are transmitted between the source and destination nodes using low-power wireless communication technologies such as ZigBee. Because they are deployed in an open environment without a security infrastructure, sensor nodes are exposed to potential threats such as memory capture by an attacker [17]. If secret information (key or control authority) of a node is stolen, an attacker can use it to inject false data into the network as a false positive attack (FPA) [18], [19], [20], or disrupt a normal report as a
The remainder of this paper is organized as follows: Section II introduces existing filtering methods for sensor networks. Section III provides the background of the proposed method and presents DEVS. Section IV gives the false negative attack (FNA) [21], [22], [23] to interfere with a service provision.

Communication methods using various security approaches to detect FPA and FNA have been studied [18], [19], [24], [25], [26], [27]. In an en-route filtering technique-based protocol, in particular, upon detecting an event, a sensor node constructs a report through interactions with neighboring nodes and transmits the report to the base station (BS) [20]. Also this approaches verify the report using the message authentication codes (MAC). When intermediate nodes receive the report, they verify it using a symmetric key-based authentication method. This approach can effectively detect attacks at a low computational cost by assessing the integrity of the report. However, these security methods guarantee only the clarity of content of the report for small node attacks, still making the methods vulnerable to massive node attacks, denial of service (DoS) attacks on legitimate data and report interruption attacks and optional forwarding attacks. A massive node attack causes abnormal behavior of the actuator by creating a false report including abnormal data. If a DoS attack on a normal report occurs, shaded regions appear in the network owing to problems such as an energy depletion of the node. Report-interruption attacks and selective forwarding attacks prevent reports from reaching the BS or convert normal reports into false reports. This type of attack can be easily detected through behavior verification. The difficulty of the existing method is that it relies on secret information to provide security, and it is difficult to recognize detailed behaviors, so behaviors cannot be verified. In order to recognize detailed behavior, the duration of the value is important. In this paper, behavior based on the history trajectory enabling the detection of an FPA and an FNA is analyzed, and further details on a false report restoring method are provided. A history trajectory refers to the recorded spatiotemporal data sets observed during state transitions of the node. The proposed method can easily calculate the duration of the state by using the history trajectory. The proposed method detects anomalous behaviors based on the history trajectory generated from the discrete event system specifications, that is, from Discrete EEvent system Specification (DEVS) based models defined for the domain process. Table 1 compares the characteristics of the SEF and CFFS protocols, which are en-route filtering methods, and the proposed method.

| TABLE 1. Comparison of sensor network security protocols. |
|-----------------|-----------------|-----------------|-----------------|
| Subject         | SEF [25]        | CFFS [19]       | Proposed        |
| FPA             | Yes (Limited)   | Yes (Limited)   | Yes             |
| Geo-based FPA   | No              | Yes             | Yes             |
| Abnormal behavior detection | No    | No              | Yes             |
| Routing path analysis | No   | No              | Yes             |
| Dynamic network security control | No   | No              | Yes             |

The statistical en-route filtering (SEF) and cluster-based false-data filtering (CFFS) methods shown in Table 1 are en-route-based false report detection approaches focusing on FPA detection. SEF and CFFS detect false reports only under limited conditions when the number of compromised nodes is smaller than the specified security threshold. Unlike traditional filtering methods, the proposed approach detects both FPAs and FNAs.

The main contributions of this study can be summarized as follows:

1) Abnormal behavior detection: Although some schemes [18], [28], [29], [30], [31] detect attack behavior using the trajectory, they solve the security problem by considering only the change in values without considering the duration time of the values on the trajectory. Therefore, it is difficult to solve problems that must consider the duration. For example, it is difficult to solve the advanced FPA and FNA attacks presented in section IV herein. The duration for each state must accurately represent the behavior. For example, as in the example part described in section IV, when a report validation node receives a report, the node creates a state transition for validation. The verified meaning of the report may vary depending on the time taken for verification. In general, when a node with a validation key receives a report, the validation time lasts longer than if it does not have the same key. These meanings analyze the behaviors through history trajectory information. Therefore, the proposed method can calculate whether the duration for each state measured through the history trajectory is within the normal range. Appropriate follow-up actions can be taken according to the analysis results of the history trajectory. As a result, the energy efficiency of a node in a WSN is improved and the legitimate report reception rate of the BS is increased.

2) Security enhancement potential: With the proposed method, the historical trajectory data generated through the execution of the DEVS model are stored and used to solve security problems. The history trajectory collected from the model is then maintained in spatiotemporal trajectory storage. The collected history trajectory is saved in the spatiotemporal trajectory storage to enable easy monitoring of the behavior by utilizing the state trajectory. To construct the history, the trajectory must be effectively extracted, which is achieved through the DEVS methodology. Such data storage can be used to solve various problems in the future, such as responses to potential threats.

3) Security framework: The proposed approach represents security knowledge as a model, thereby providing security scalability in detecting attacks. In addition, security intelligence can be enhanced by defining new security knowledge as a security model and then storing the model in the proposed system. Therefore, the proposed structure can be easily used as a secure system framework in various domains.
problem statements and a detailed description of the proposed method. Section V presents a performance evaluation of the proposed method based on analyses and simulations. Finally, Section VI offers a discussion of the security knowledge as well as some concluding remarks regarding this research.

II. RELATED WORK
In a WSN, sensor nodes are vulnerable to an FPA and FNA in the application layer owing to their limited energy resources [32]. Application layer attacks consume unnecessary communication energy from intermediate nodes and block normal event content from being transmitted. SEF detects false reports through the validation phase using a shared global key pool [25]. However, this scheme does not allow the BS and intermediate nodes to detect false reports when many compromised nodes participate in the report generation. If a false report reaches the endpoint, report-based applications can malfunction. It is therefore essential to filter false reports under an extensively compromised situation. CFFS groups nodes into clusters and constructs a tree structure for false report detection through cooperative verification between nodes [19]. This scheme detects false reports through intermediate filters even if nodes placed in geographically different locations are compromised. However, if a compromised node attempts a replay attack, the energy from the source node to the intermediate filtering node is quickly depleted [32]. The detection and blocking of damaged nodes is important because energy depletion on the upstream nodes can cause the disconnection of a local network. In addition, various en-route filtering schemes have been proposed, including IHA [24] and DEF [26].

The en-route filtering approach is effective when a node is compromised at a small scale but is difficult to detect when a node is massively compromised. Large-scale attacks participate in many compromised nodes, generating strong false reports. Such an attack can also tamper with event contents in normal reports. Strong false reports neutralize existing security schemes and cannot be detected by verification nodes or a BS. It is therefore important to detect massive node attacks.

We propose an abnormal behavior detection scheme using a monitoring system based on the history trajectory. The history trajectory stores attack and defense knowledge in a spatiotemporal form, and thus it can effectively detect incorrect behavior. In addition, the history trajectory is used as a new module within a DEVS simulation. Therefore, this module detects incorrect behavior based on the analysis results of the DEVS simulation.

III. BACKGROUND
This section introduces the security methods provided through a WSN application and describes the DEVS formalism, which is the basis of the proposed method.

A. WSN APPLICATION BEHAVIOR DETECTION METHODS
A WSN refers to an environment in which sensor nodes monitor environmental conditions such as temperature, pressure, and sound, thereby forming a network between sensor nodes for transmitting data. In this environment, two-way communication enabling both sensor control and monitoring is possible; thus, decisions are made based on large-scale values measured in a WSN. Recent research has focused on the development of applications that are sensitive to the collected data, such as control applications through the monitoring of the machine work processes [33], [34]. In the field of disaster response, many applications based on WSN network technology are becoming increasingly popular [35].

B. DEVS FORMALISM
The DEVS formalism proposed by Zeigler provides a means of specifying mathematical entities called systems [36], [37].

The DEVS formalism is divided into two model types, i.e., a basic model and a multi-component model, whereby the behavior and structure are described by specifying a target system in the real world.

The basic models specified by DEVS formalism can be defined through the following structure:

$$M = (X, S, Y, \delta_{ext}, \delta_{int}, \lambda, t_a). \quad (1)$$

where $X$ denotes a set of external input ports through which events are received, $S$ is a set of state variables, $Y$ is a set of external output ports through which events are output, $\delta_{ext}$ is an external transition function that specifies how the system changes state when an input is received as an external port, and $\delta_{int}$ is an internal transition function. If the time elapses according to the $t_a$ function of the system, the model system is $\delta_{int}$, which changes its state independently; in addition, $\lambda$ is an output function that produces an external output before $\delta_{int}$ occurs, and $t_a$ is a time advance function [36], [37].

$$N = (X, Y, D, [M_d]_{d \in D}, \text{EIC}, \text{EOC}, \text{IC, Select}). \quad (2)$$

In the above formula, $X$ and $Y$ are the same as $X$ and $Y$ of the basic model, $D$ is the set of component names for each $d$ of $D$, and $M_d$ is a basic or coupled DEVS model. The coupled model is atomic or a model other than itself. The model structure combined with $N$ represents a new component model. EIC is a set of external input couplings that connect external inputs to the component model inputs, EOC is a set of external output couplings that connect external outputs to the component model outputs, and IC is a set of internal couplings that connect the component outputs to the component inputs for selecting a tie-breaking function [36].

IV. PROPOSED METHOD
In this section, the threat model, which is the goal of this thesis, is described. An overview of the proposed method is illustrated with the system entity structure (SES) representing the network structure to which the proposed method is
applied. In addition, further details on the proposed method are provided and the effectiveness of the proposed method is introduced based on four example attack scenarios.

A. PROBLEM DEFINITION

Most of the MAC-based security methods are used to efficiently verify reports in a WSN. Nodes create MACs using shared keys and event content. Nodes are easily compromised by attackers because they are deployed in an open environment. Attackers can attempt to conduct an FPA or an FNA by exploiting the nodes. The vulnerabilities found in a WSN are as follows:

- False data and MAC injection attack as an FPA
  With this attack, the attacker compromises the sensor nodes through a potential attack (physical or software) and uses it to inject false reports into the network [38]. False reports induce the BS to conduct an unnecessary response, causing increased network traffic congestion, false alarms, and a waste of node resources.

- Massive node compromise
  The parameter security threshold in en-route filtering methods trades false data detection power for overhead. In [18], [19], [24], [25], [26], and [27], a systematic response was shown when the node below the security threshold was under a compromised condition. This attack was beyond the security threshold and created a strong false report that even the BS could not detect. Such strong false reports can lead to serious accidents in IoT applications [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14].

- Report disruption attack as an FNA
  With this attack, the attacker intentionally injects false MACs or invalid data into the CH node [21], [22], [23]. When a CH node enters the report generation phase, these false reports obstruct the report transmission by filtering the data on the intermediate nodes.

The main attacks from compromised nodes are as follows:

- A compromised node intentionally deletes the message it receives.
- The compromised node collects the transmitted message and sends it to another node after a forgery.
- A compromised node changes the detected value to another abnormal value.
- The compromised node changes the routing path and sends a report to a different node than the intended one.

B. GOAL

- The BS collects all data detected in the network through additional node arrangements such as CAA-PVFS, or through additional information in the transmitted packets. That is, the proposed system can record the history trajectory as a model.

- With the proposed method, a node is considered compromised when it conducts an action that is not defined in the behavior knowledge of the trajectory analysis of the sensor node. This is achieved by monitoring the continuous state of the node. The proposed method can block injected data from that node or restore corrupted reports.

- In this way, it is possible to prevent the accumulation of false data from sensor nodes and provide normal IoT services.

C. PROPOSED METHOD OVERVIEW

The proposed method is applied in three phases: a state message collection phase, state inference phase, and response phase. In the state message collection phase, nodes send their current state as a message to the history trajectory-based detection (HTBD). This phase is achieved when the states of the nodes transition. In [18], a context aware architecture, which replicates the behavior of actual systems, was proposed to monitor real systems. Using this method, the data collection system can synchronize the model using a device in the real world. When the HTBD receives a state message, it conducts a state transition by mapping it to the model defined in the DEVS.

This allows the HTBD to obtain a trajectory for the actual node. The collected trajectory is used in the state inference phase. The state inference phase is executed when a message created in the state collection phase is received or an anomaly
is found. During this phase, the trajectories stored as security knowledge are compared with the collected trajectories. If the collected trajectory differs from the normal trajectory, HTBD finds nodes with abnormal behavior based on the security knowledge. During the response phase, the node with abnormal behavior is added to the blacklist table and broadcast to the sensor network. A node that receives this message can either reject the message by adding the node to the blacklist or update the routing path.

Fig. 1 shows the reasoning process of the proposed method. The sensor field on the left of Fig. 1 represents the sensor network deployed in the real world, whereas the one on the right represents the structure of the proposed method with HTBD knowledge base. The HTBD consists of an entity structure base, spatiotemporal history data, and a model base. The entity structure base expresses the structure of the sensor field based on the DEVS expert system format. The spatiotemporal history data are represented in a state timing diagram that expresses the flow diagram of the states of the models over time. In the real world, the model base is a set of models created based on the DEVS formalism for components such as the CH node, member node (MN), and BS in the sensor field.

D. SYSTEM ENTITY STRUCTURE (SES)
Fig. 2 shows the structural diagram of the simulation model for the proposed method based on the WSN-SES [39]. This structural diagram has the advantage of being able to easily switch to a structural model by expressing the system as a graphical user interface and defining the structural relationships [40]. Through this process, the structure of the system can be expressed hierarchically. The FPA and FNA models represent attack scenarios, and the WSNGENER model represents models for a general event occurrence. TRANS and CLUSTER_TRANS models are used for a performance analysis by examining the collection of activity reports. BS, MN, and CH are models that imitate objects existing in the real world and apply the following actions.

1) SPATIOTEMPORAL CH MODEL
CH is the model corresponding to the CH node of the sensor network. In this model, the report is generated by receiving the MAC generated from the MN models, and the report is verified by the report received from the CH models. During the verification phase, the behavior of the CH is analyzed using the security knowledge defined in the model in a spatiotemporal form.

2) SPATIOTEMPORAL MN MODEL
When an MN detects an event within the sensor network, it generates a MAC based on the detected content as well as its own security key to transmit the MAC to the corresponding CH node.

3) SPATIOTEMPORAL BS MODEL
The BS model acts as a collection agency for the report and analysis of the history trajectory. The proposed BS model analyzes the trajectory history data stored therein to find the compromised node when the data result in applications that are determined to be incorrect. If a compromised node is found through the inference process, the BS notifies the CH node adjacent to the compromised node, blocks the data generated at that node, and finds a new path for rerouting.

E. DEVS-BASED HISTORY TRAJECTORY
We implemented the HTBD through simulation models using DEVS. For the HTBD, we implemented a spatiotemporal trajectory storage using information obtained from the SEF and CFFS to effectively identify compromised nodes in the sensor network.

The proposed method can determine the intrusion process by recognizing the situation based on the history trajectory. The model can maintain a secure WSN by effectively responding to incorrect actions. These WSN systems consist of various monitoring sensors and semiconductor devices that are combined with signal converter processing chips. The monitoring system starts with these sensor nodes. The sensors are initially responsible for collecting data from the real world, after which they transform and process the raw data into the digital realm. A sensor node has various defined states and reports the status to the BS according to the actual status behavior. Status collection is not addressed herein because it is outside the scope of the present study. In the HTBD, the states of the nodes can be collected using various data collection methods to apply state transitions and conduct discrete event simulations, as shown in Fig. 3.

Fig. 3 shows the variation in history trajectory over time, which fluctuates in real time as the state of the CH node is collected from the BS. As shown, the HTBD can observe the activities of each node (t0 through t12), and in this way can analyze whether the behavior is normal. Whether the node state changes to a general pattern within a normal time range can also be determined. In the state transition diagram of the CH model shown in the upper-left corner of Fig. 3, 10 states...
are defined. In $S_0$, $S_2$, and $S_5$, a branch exists for the next state. These branches represent determining the subsequent state according to the previous trajectory state. A solid line means a general state transition, and a dotted line means a state transition when an attack is underway. If a pattern of $t_4$ to $t_7$ is introduced, traffic data of other models are examined, and a plan is implemented according to the results. The model returns to the normal trajectory if the response is normal, as shown in $t_8$ through $t_{10}$. Because model-based trajectory tracking stores all actions of the models, it is possible to identify the signs as well as the attack process. Because the BS collects and stores the states for all nodes, it can be easily used to analyze the historical data of multiple nodes as needed.

Algorithm 1 shows the HTBD protocol for the state verification as a pseudocode when the BS receives an event report or state message. The report verification process first checks whether the number of MACs in the report is equal to the set threshold. If the two thresholds differ, the actions are analyzed up to the node that is the source of the event by first analyzing the last node that has a change in state in the history trajectory. If a node creates or applies a MAC generation multiple times, it is regarded as an FPA and that node is blocked. If the behavior of the MN node is normal and the CH verification procedure is also normal, it finds the node that replaces the contents of the report, considers the node as an FNA and blocks it, while resetting the routing path of the damaged node and those nodes related to routing. In addition, the contents of the report are considered normal and are saved. If an intermediate node detects a false MAC and is determined to be an FNA attack, a new MAC is created using the event contents in the corrupted report and its own key. Moreover, the corrupted MAC in the false report is replaced with the newly created MAC.

If the threshold of the compromised node is greater than the security threshold, the nodes participating in the report are identical at each time point unless a large number of nodes are compromised. The HTBD analyzes the behavior of the MN node and determines that it is damaged by more than T if the node participating in the report is repetitive. Follow-up measures, such as adjusting the security threshold or excluding the area from the monitoring area, are then taken. By applying this process, the HTBD can distinguish between an FNA and an FPA, prevent the injection of false data, and determine whether an intrusion has occurred. The proposed method can therefore be considered robust to report tampering.

F. EXAMPLES
This section discusses examples of FPAs and FNAs. In one sensor field, all nodes are deployed with key assignments and validation nodes are selected. For clarity, an example of the proposed method is shown in Fig. 4.

1) ATTACK CASE 1 (FPA - SOURCE NODE COMPROMISED CASE)
Attack case 1 represents a situation in which FPA occurs when a source node is compromised in a sensor field. Fig 5 depicts the detection process for the infringing node in detail based on the spatiotemporal history data. An attacker creates a MAC using compromised nodes to inject a report containing a false event into the network. In addition, a false report is generated using the previously generated fake MAC and randomly modified MAC. An attacker can generate false reports by generating randomly modified MACs using the specified security threshold value, that is, the number of compromised nodes. An attacker must match the number of MACs included in the report to the security threshold value.
Algorithm 1 HTBD Protocol

1: procedure HTBD.receive(Mp)
   //Mp is a phase value of the DEVS component model
2: transition M set Mp
3: if M is a DEVS component model
4: update Mp //Mp is history trajectory of M
5: if Mp ≠ M or Mcp = ‘error’ then
   //Mcp is current phase of M
6: HTBD.detection()
7: end if
8: end procedure
9: procedure HTBD.receive(R)
10: //R is the event report generated by a CH node
11: if T ≠ R.ML then //R.ML is MAC length of R
12: //Security threshold verification phase for R
13: HTBD.detection()
14: end if
15: while idx of MACs
16: if Kgkp.idx = every MAC.idx then
17: //Kgkp is the key of the global key pool stored in BS
18: //Kgkp is_MACs detected as FPA
19: else if n.key ≠ MAC.idx then
20: n.key = the key index of the forwarding node n
21: add BLFPA = {Fn}
22: //BLFPA is a blacklist detected as FPA
23: else if n.key ≠ MAC.idx then
24: add BLFNA = {Fn}
25: //BLFNA is a blacklist detected as FNA
26: end if
27: while
28: end procedure
29: procedure HTBD.detection()
30: while Fn of all intermediate nodes for report transmission
31: set M = Fn
32: if Mt ≠ FNAo then //FNAo is a normal pattern of forwarding node n
33: add BLFPA = {Fn}
34: //BLFPA is a blacklist detected as FPA
35: else
36: //MNKo is a normal pattern of the sink node
37: add BLFNA = {Fn}
38: //BLFNA is a blacklist detected as FNA
39: end if
40: end while
41: end procedure
42: if Mt = 0 then
43: if Mt ≠ MNKo then //MNKo is a normal pattern of member node n in cluster
44: send {blacklist [n], ‘drop’}
45: //Block compromised nodes defined in the blacklist
46: end if
47: if Mt ≠ MNKo then //MNKo is a normal pattern of member node n in cluster
48: send {blacklist [n], ‘restore’}
49: //Attempts to restore when a report is received from a compromised node defined in the blacklist
50: end if
51: add BL’ = BLFPA
52: set BLFNA = ‘()
53: add BL’ = BLFPA
54: set BLFNA = ‘()
55: //Initialize FNA and FPA blacklists
56: end procedure

FIGURE 4. Four attack examples of intrusion prevention.

to prevent false reports from being filtered. Therefore, one node generates multiple MACs. The report, including the generated false MAC, is transmitted to the BS through the forwarding nodes. In a false report, at the verification stage of the BS, the MAC generated in the report may be detected as false. Once the report is identified as false, the HTBD determines whether it is an FNA or an FPA using the history data, as shown in Fig. 5. In the case of an FPA, the suspicious act of generating multiple MACs at one node is applied, and the CH receives an abnormally large number of MACs. As shown in the intermediate node history trajectory of Fig. 5, the contents of the neighboring nodes of the source CH node, which is the source of the false report, are analyzed. If the number of neighboring nodes that generated a MAC is below the security threshold value and multiple MACs are generated on one node, the nodes that participated in creating the false reports are determined to be compromised nodes. With the HTBD, a compromised node informs other neighboring nodes of the source for reconstructing the routing path and adds a list of compromised nodes to the blacklist to block the reception of counterfeit data from that node.

2) ATTACK CASE 2 (FPA - MASSIVE SOURCE NODE COMPROMISED CASE)

Attack case 2 represents an attack environment in which the number of compromised nodes exceeds the security threshold
value. As described in attack case 1, a MAC above the security threshold is required to generate a strong false report. The compromised nodes generate an arbitrary event for generating MACs. Generated MACs can be used to create strong false reports. Strong false reports avoid verification of the intermediate nodes and BS.

When the security threshold is exceeded, the verification node and BS cannot detect false MACs at a rate corresponding to the verification. Such event data may be executed in IoT or other application programs. However, because the HTBD has time data, false data can be identified by identifying features that consider the geographical characteristics of the sensor nodes. For example, if the processing times required for nodes to create MACs within a cluster are the same, the CH node will be the first to receive a MAC generated from a node that is close to it. However, it is difficult for an attacker to determine this pattern. If the HTBD defines an event content occurrence behavior that is suitable for the domain of a model, the normal event occurrence order trajectory can be known. As a result, when the HTBD detects an incorrect event input, it determines that the source cluster for false reports has been compromised and increases the level of security for that cluster.

3) ATTACK CASE 3 (FNA - SOURCE NODE COMPROMISED CASE)
In attack case 3, the source node of the event is compromised. When an event is detected, the CH node transmits the event occurrence message to the member nodes. After checking whether the received event data match the content of the received event data, a MAC is generated and transmitted to the CH node. In addition, the compromised nodes that receive the event data generate a false MAC and transmit it to the CH node. The CH node then collects the MACs and creates a false report. False reports are detected through the en-route or BS verification process.

At the same time, with the HTBD, the process from sensing the event in the source node to filtering is analyzed using the history data. However, if all processes are recognized normally, the event report is determined to be normal, and the node that generated the detected false MAC is determined to be a compromised node. Thereafter, the intermediate node replaces and transmits the false MAC in the report with the MAC that it generates. If the false report reaches the BS, the BS can find the false MAC using the global key pool. Likewise, if all actions are determined to be normal when analyzing the trajectory log of the node, the node that generated the false MAC is determined to be a compromised node, and the corresponding ID is notified to the CH for blocking the data injection.

4) ATTACK CASE 4 (FNA - INTERMEDIATE NODE COMPROMISED CASE)
In attack case 4, an intermediate compromised node attempts an FNA. If a forwarding node located in the middle of a routing pass is compromised, the report drops the normal
and 180 MNs) for a WSN. The experimental environment sensor field consisted of a BS and 200 sensor nodes (20 CHs the proposed method and compare it to SEF and CFFS. The simulation. A simulation experiment was conducted to evaluate the behavior of real objects can be expressed in the same abstracting the functions of the real-world objects, and thus software based on DEVS. The DEVS model was created by previous source sensor node to resend the report.

| Parameter               | Value          |
|-------------------------|----------------|
| Number of nodes         | 200            |
| Network field size      | 400 × 40 m²    |
| Base station location   | 400 × 20 m     |
| Number of clusters      | 20             |
| Number of member nodes in a cluster | 10          |
| Security threshold      | 3, 5, 7        |
| Compromised ratio       | 50%            |
| Size of report          | 36 bytes       |
| Size of MAC             | 1 byte         |
| State message size      | 2 bytes        |
| Generate MAC            | 15 µJ          |
| Verify report           | 75 µJ          |
| Transmit report         | 16.25 µJ/byte  |
| Receive report          | 12.5 µJ/byte   |

The HTBD monitors whether nodes change the process of receiving, verifying, and transmitting reports in a normal pattern according to the status of nodes collected in real time. If an attack occurs, report filtering proceeds at the BS or verification node, where the history data of the HTBD are analyzed. If the analysis results display a behavior other than the verification pattern at the intermediate node, a routing path bypassing the compromised node is reconstructed. Various studies have been conducted on route reconfigurations. When a new routing pass is formed, a report request message is sent to the previous source sensor node to resend the report.

V. EXPERIMENT

A. EXPERIMENTAL ENVIRONMENT

We conducted a modeling and simulation using DEVS-ObjC software based on DEVS. The DEVS model was created by abstracting the functions of the real-world objects, and thus the behavior of real objects can be expressed in the same way. Table 2 shows the network properties for the simulation. A simulation experiment was conducted to evaluate the proposed method and compare it to SEF and CFFS. The sensor field consisted of a BS and 200 sensor nodes (20 CHs and 180 MNs) for a WSN. The experimental environment of the field was 400 × 40 m² in size. These nodes were randomly and uniformly distributed on a cluster with a size of 20 × 20 m². Each cluster consisted of one CH and nine MNs; the MNs joined the CH within one hop. Routing paths in the network were established based on the Dijkstra shortest path algorithm and two-tier data dissemination scheme [41], [42]. The CHs were reported to the BS in the forward direction through multiple hops, and the BS collected additional state reports. The transmission range of the node was written according to MICAz mote. The BS was located in the upper-right side of the sensor field. The energy consumption of the SEF was used for the experiment. Each node employed 16.25 µJ per byte for transmission, 12.5 µJ per byte for reception, and 15 µJ per byte for generation. To verify the MAC in the intermediate node, the node consumed 75 µJ [25], [43]. The size of a report was 36 bytes. In addition, four specific clusters were randomly selected to generate four types of attack, and the compromised nodes were placed in those clusters. The compromised nodes generated an attack according to the false traffic ratio (FTR). In the simulation experiment, 100 events were generated at random locations.

B. EXPERIMENTAL RESULTS

Fig. 6 shows the energy consumption of the network when the FTR was 0%–100%. As the FTR increased, the proposed method blocked the node and took follow-up measures to adjust the security threshold. Therefore, in comparison to SEF, the overall energy consumption when applying the proposed method decreased by 16.150%, 43.618%, and 63.420% when the total energy consumption was 20%, 40%, and 60%, respectively. However, existing approaches such as SEF show a higher energy consumption than the proposed method because their detection is based on the MAC and geographic location. However, when the attack rate was 0%, the energy consumption of the proposed method was 6.402% higher because the proposed technique saved the history data. Therefore, whenever the nodes changed their state, a report was sent to the BS. However, the methods shown in Figs. 6 (a), (c), and (e) were found to reduce the energy consumption by 46.737% in comparison to the SEF and by 41.927% in comparison to CFFS through early detection using the history trajectory. When the number of compromised nodes was larger than the security threshold, because SEF and CFFS could not detect any abnormal behaviors for cases (a) and (c), a similar energy consumption was shown in a clean environment. However, the proposed method effectively dropped the false report generated from incorrect abnormal patterns through a history trajectory analysis. As shown in Figs. 6 (c) and (d), the CFFS achieved an efficient filtering with the appropriate security strength settings. Because cases (e) and (f) in Fig. 6 were set to a high-security threshold, their early detection ratio was also high.

Fig. 7 shows the total energy consumed versus the FTR during an FNA. With the proposed method, the middle node restores the MAC of the report, and the report reaches the BS normally. This indicates that as the total number of hops for
the report increases, the proposed method has a higher average consumption rate than the existing approach. When an FNA attack occurs, the filtering hops are the same as with the existing method. Even if the node above the security threshold is damaged, the HTBD can detect the attack because it has attack knowledge through the history trajectory.

In an FNA, when a normal report does not reach the BS, the number of successful attacks increases. The number of successful attacks with the proposed method was maintained at zero, whereas the number increased gradually with the attack ratio for the SEF and CFFS methods. Because the proposed method restores the false report using the security...
knowledge of the FNA, the number of successful attacks is effectively reduced, as shown in Fig. 8. Unlike with the FPAs, as Fig. 8 indicates, the FNAs showed the same number of successful attacks regardless of the security threshold.

VI. FURTHER DISCUSSIONS

The main purpose of the proposed scheme is to detect advanced attacks that occur in WSN and prevent abnormal behavior. Traditional approaches provide security through mathematical encryption. However, this approach can’t detect attacks if the node has been compromised extensively. The proposed scheme improves security by detecting compromised nodes based on the trajectory regardless of the number of the compromised node.

The proposed method detects compromised nodes by generating trajectories through the DEVS models. generating trajectories method was achieved through a DEVS formalism based on the security model, history trajectory base, and abstract simulator, as described in Section III. The DEVS models with a 1:1 match in the real world can potentially have historical data consisting of spatiotemporal information. The proposed method recognizes the behavior of the model through this historical data. Security knowledge is an important factor in determining whether the actions described above are legitimate. In the proposed method, security knowledge of high-level security capabilities is defined as atomic models. We conducted experiments on four examples as introduced in section IV and compared the CFFS and SEF protocols and proposed techniques as shown in Fig. 6. As a result of the experiment, more energy is consumed in the proposed method due to the overhead of sending and receiving messages to collect the status of nodes. In pure environments such as FTR 0% in Fig. 6, the proposed method may appear to have a shorter lifetime. However, considering the attack situation, the proposed method is more efficient because it enables a detailed analysis of network behavior through state collection. In the spatio-temporal history data storage of the proposed HTBD system, only the node attack trajectory for attack detection and the node and en-route filtering method normal trajectory for abnormal behavior prevention is stored. If a new attack type is discovered, the attack can be easily detected if only new attack knowledge is added to the storage module. In this approach, the analysis module and the storage module are divided, so that the network security can be easily managed. In future research, we plan to apply spatio-temporal database to increase the efficiency of the storage module.

VII. CONCLUSION AND FUTURE WORK

In a WSN, nodes are distributed in an open environment; thus, it is easy for attackers to attempt an FNA or an FPA by damaging the nodes. To solve this problem, report verification methods based on symmetric keys have been studied. However, traditional methods have a problem in that normal reports may be recognized as false reports when an FNA occurs. In this study, security was provided by recognizing the situation and identifying and analyzing the behaviors of the WSN components. In particular, actions against FNAs and FPAs are defined in the models with respect to two factors, i.e., the time and state spaces, through which the model trajectory is formed to detect an attack. Because attack patterns are formed differently depending on the operation process of the models, the proposed technique classifies attacks according to the pattern and initiates an appropriate response according to the attack. Based on the experimental results obtained, it was confirmed that the proposed method increases the network lifetime and reduces the number of detection hops for false reports in comparison to other previous approaches.

To establish a high-level intelligent security network, it is necessary to manage the security knowledge [44], [45]. Therefore, as a future study, we plan to construct an intelligent security system that can efficiently manage knowledge using a temporal database with security based on BM-DEVS [46].

REFERENCES

[1] M. P. Durišić, Z. Tafa, G. Dimić, and V. Mihalić, “A survey of military applications of wireless sensor networks,” in Proc. Medit. Conf. Embedded Comput. (MECO), Jun. 2012, pp. 196–199.
[2] H. Alemdar and C. Ersoy, “Wireless sensor networks for healthcare: A survey,” Comput. Netw., vol. 54, no. 15, pp. 2688–2710, Oct. 2010.
[3] T. Ojha, S. Misra, and N. S. Raghuvanshi, “Wireless sensor networks for agriculture: The state-of-the-art in practice and future challenges,” Comput. Electron. Agric., vol. 118, pp. 66–84, Oct. 2015.
[4] Y. Liu and J. Jiang, “Application of wireless sensor network for monitoring water temperature difference in blast furnace cooling system,” in Advanced Electrical and Electronics Engineering. Berlin, Germany: Springer, 2011, pp. 409–414.
[5] B. Rashid and M. H. Rehmani, “Applications of wireless sensor networks for urban areas: A survey,” J. Netw. Comput. Appl., vol. 60, pp. 192–219, Jan. 2016.

[6] F. Losilla, A.-J. Garcia-Sanchez, F. Garcia-Sanchez, J. Garcia-Haro, and Z. J. Haas, “A comprehensive approach to WSN-based ITS applications: A survey,” Sensors, vol. 11, no. 11, pp. 10220–10265, Oct. 2011.

[7] G. Fletcher, L. Xu, A. Nayak, and I. Stojmenovic, “Randomized robot-assisted relocation of sensors for coverage repair in wireless sensor networks,” in Proc. IEEE 72nd Veh. Technol. Conf. (Fall), Ottawa, ON, Canada, vol. 9, Sep. 2010, pp. 1–5.

[8] D. Kandris, C. Nakas, D. Vomvás, and G. Koulouras, “Applications of WSN-based IoT network: An up-to-date survey,” Appl. Syst. Innov., vol. 3, no. 1, p. 14, Feb. 2020.

[9] A. Benayache, A. Bilami, S. Barkat, P. Lorenz, and H. Taleb, “M3M: A microservice middleware for smart WSN-based IoT application,” J. Netw. Comput. Appl., vol. 144, pp. 138–154, Oct. 2019.

[10] H. Zhang and S. Yang, “Research on application of wsn in cold chain logistics’ warehousing and transportation,” in Proc. China Academic Conf. Printing Packag. Media Technol. (PPMT), Xi’an, China, vol. 25, 27 Nov. 2016, pp. 589–598.

[11] H. Li and A. V. Savkin, “Wireless sensor network based navigation of micro flying robots in the industrial Internet of Things,” IEEE Trans. Ind. Informat., vol. 14, no. 8, pp. 3524–3533, Aug. 2018.

[12] P. Lavanya and R. Sudha, “A study on WSN based IoT application in agriculture,” in Proc. 3rd Int. Conf. Comput. Electron. Syst. (ICCES), Oct. 2018, pp. 1046–1054.

[13] M. T. Lazarescu, “Design of a WSN platform for long-term environmental monitoring for IoT applications,” IEEE Emerg. Sel. Topics Circuits Syst., vol. 1, no. 1, pp. 45–54, Mar. 2013.

[14] A. R. Juladi, K. Khithani, P. Pawar, and K. Malvi, “Environmental monitoring using wireless sensor networks (WSN) based on IoT,” Int. J. Eng. Technol., vol. 4, no. 1, pp. 1371–1378, 2017.

[15] I. Khan, F. Belqasmi, R. Ghiitho, N. Crespi, M. Morrow, and P. Polakos, “Wireless sensor network virtualization: A survey,” IEEE Commun. Surv. Tuts., vol. 18, no. 1, pp. 553–576, 1st Quart., 2015.

[16] M. Cerchetti, F. Luti, A. Mecocci, S. Parrasso, G. Peruzzi, and A. Pozzebon, “A low power IoT sensor node architecture for waste management within smart cities context,” Sensors, vol. 18, no. 4, p. 1282, Apr. 2018.

[17] M. V. Bhartari, R. C. Tantuguri, C. Jayakumar, and K. Selvamani, “Node capture attack in wireless sensor network: A survey,” in Proc. IEEE Int. Conf. Comput. Intell. Comput. Res., Dec. 2012, pp. 1–3.

[18] M. N. Su and T. H. Cho, “Context-aware architecture for probabilistic voting-based filtering scheme in sensor networks,” IEEE Trans. Mobile Comput., vol. 16, no. 99, pp. 2751–2763, Oct. 2017.

[19] Z. Liu, J. Wang, S. Zhang, and H. Liu, “A cluster-based false data filtering scheme in wireless sensor networks,” Adhoc Sensor Wireless Netw., vol. 23, no. 1–2, pp. 21–45, 2014.

[20] A. Kumar and A. R. Pais, “En-route filtering techniques in wireless sensor networks: A survey,” Wireless Pers. Commun., vol. 96, no. 1, pp. 697–739, Sep. 2017.

[21] H. Y. Lee and T. H. Cho, “A scheme for adaptively counteracting application layer security attacks in wireless sensor networks,” IEICE Trans. Commun., vol. 93, no. 7, pp. 1881–1889, 2010.

[22] A. Al-Riyami, N. Zhang, and J. Keane, “An adaptive early node compromise detection scheme for hierarchical WSNs,” IEEE Access, vol. 4, pp. 4183–4206, 2016.

[23] S. Oroum, B. Kantarci, and H. T. Moutfah, “Mitigating false negative intruder decisions in WSN-based smart grid monitoring,” in Proc. 13th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC), Jun. 2017, pp. 153–158.

[24] S. Zhu, S. Setia, S. Jajodia, and P. Ning, “An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks,” in Proc. IEEE Symp. Secure Privacy, May 2004, pp. 259–271.

[25] F. Ye, H. Luo, S. Liu, and L. Zhang, “Statistical en-route filtering of injected false data in sensor networks,” IEEE J. Sel. Areas Commun., vol. 23, no. 4, pp. 839–850, Apr. 2005.

[26] Z. Yu and Y. Guan, “A dynamic en-route filtering scheme for data reporting in wireless sensor networks,” IEEE/ACM Trans. Netw., vol. 18, no. 1, pp. 150–163, Feb. 2010.

[27] R. Lu, X. Lin, H. Zhu, X. Liang, and X. Shen, “BECAN: A bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 1, pp. 32–43, Jan. 2012.

[28] Y. Ji, L. Wang, W. Wu, H. Shao, and Y. Feng, “A method for LSTM-based trajectory modeling and abnormal trajectory detection,” IEEE Access, vol. 8, pp. 104063–104073, 2020.