entropy

MDPI

Article

Intelligent Security IT System for Detecting Intruders Based on Received Signal Strength Indicators

Yunsick Sung

Faculty of Computer Engineering, Keimyung University, Daegu 42601, Korea; yunsick@kmu.ac.kr

Academic Editor: James J. Park
Received: 10 September 2016; Accepted: 10 October 2016; Published: 16 October 2016

Abstract: Given that entropy-based IT technology has been applied in homes, office buildings and elsewhere for IT security systems, diverse kinds of intelligent services are currently provided. In particular, IT security systems have become more robust and varied. However, access control systems still depend on tags held by building entrants. Since tags can be obtained by intruders, an approach to counter the disadvantages of tags is required. For example, it is possible to track the movement of tags in intelligent buildings in order to detect intruders. Therefore, each tag owner can be judged by analyzing the movements of their tags. This paper proposes a security approach based on the received signal strength indicators (RSSIs) of beacon-based tags to detect intruders. The normal RSSI patterns of moving entrants are obtained and analyzed. Intruders can be detected when abnormal RSSIs are measured in comparison to normal RSSI patterns. In the experiments, one normal and one abnormal scenario are defined for collecting the RSSIs of a Bluetooth-based beacon in order to validate the proposed method. When the RSSIs of both scenarios are compared to pre-collected RSSIs, the RSSIs of the abnormal scenario are about 61% more different compared to the RSSIs of the normal scenario. Therefore, intruders in buildings can be detected by considering RSSI differences.

Keywords: security; beacon; Bayesian probability; localization; RSSI

1. Introduction

Given that entropy is the key concept of IT security systems, diverse kinds of security algorithms based on entropy have been developed and applied. Currently, for security, iris recognition [1], fingerprint recognition [2], and other methods are utilized. Tags are one of the broadly utilized approaches in security IT systems. Each tag is assigned by the corresponding authority and allows authorized employees to move within allowed areas. However, tags have the critical weakness that they can be held by unauthorized employees. Tags can be replaced by other security systems, but at significantly greater cost.

For tag utilization, their weakness should be addressed. First, when unauthorized persons utilize tags, the IT security system should alert of an unauthorized situation. To do so, approaches to recognize the owners of tags should be provided. The owners can be recognized by the patterns of utilization of their tags, but the weakness of the assessment of tag utilization patterns is that the utilization patterns of authorized persons can be similar to those of unauthorized persons. Therefore, a novel approach that classifies the two types of patterns is required. For example, additional environmental information can be utilized such as the time when tags are utilized or the movement patterns of owners who hold tags.

This paper proposes an intelligent intruder detection security approach based on received signal strength indicators (RSSIs) of tags collected by access points. For example, the RSSIs of tags are measured and analyzed to recognize the patterns of the movements of residents. Abnormal patterns

are detected by comparison with pre-collected normal patterns of the RSSIs. Therefore, illegal intruders can be detected from their abnormal patterns.

The rest of this paper is structured as follows. Section 2 introduces security-related research and demonstrations-based learning research. In addition, the requirements for intruder-detecting IT systems are suggested. Section 3 describes the concept of the proposed method and then describes the proposed method in detail. Section 4 shows the results of the proposed method from experiments. Finally, Section 5 presents conclusions.

2. Related Work

In this paper, wireless signal-related research and demonstration-based learning research are combined to propose a novel intelligent IT security system. This section introduces both the research and the requirements of intelligent IT security systems to detect intruders.

2.1. Wireless Signal-Related Research

While the global positioning system (GPS) is utilized to find outdoor locations, multiple approaches are utilized for indoor localization. Wireless signal-based localization is one of the core approaches used [3]. For example, crowd-sourced mapping and localization (CMAL) is a wireless-fidelity (WiFi) localization system based on simultaneous localization and mapping (SLAM) [4]. A CMAL client generates a WiFi-based global finger print map by utilizing multiple smart phones and a CMAL server manages the global finger print map. Research has been done that utilizes a dynamic Bayesian network (DBN) and generates a probability graph to estimate indoor locations [5]. As we described, it is difficult to find the locations in indoor environments. DBN-based research estimates indoor locations based on the generated probability graph. In addition, particle filter is also proposed to trace indoor locations approximately. Finally, the quality of collected WiFi-based signals is improved by Gaussian interpolation [6].

Beacon-based signals are also utilized for indoor localization. For example, research on localization system using a 3D triangulation algorithm has been introduced [7]. 3D triangulation algorithm is a method of calculating a location from three points and three measured three distances from these points. Therefore, this research finds a currrent indoor location based on three beacon-based wireless signals.

2.2. Demonstration-Based Learning Research

For analyzing RSSI patterns, demonstration-based learning approaches as described below can be applied to IT security systems. Motor primitives executed by autonomous robots consist of consecutive movements. Prior research on the learning of motor primitives using demonstration-based learning suggests that the configuration of approaches based on differing learning goals is possible, including learning the actions of virtual agents in a virtual environment [8] and learning motor primitives of robots in a real environment [9,10]. The previous demonstration-based learning research is introduced and the problems encountered with learning processes are described below. Finally, we compare previous work with our proposed approach.

One focus of research has been learning motor primitives by observing and then dividing consecutive predecessor movements [11]. In this method, the points of division between consecutive movements are determined using a cost function to calculate variance. However, sections where division points are frequently made result in multiple motor primitives with short execution times; conversely, sections with fewer division points owing to smaller variance result in motor primitives with excessively long execution times.

The progress of motor primitive generation can also be facilitated in the field of vision systems by applying methods for observing and imitating motor primitives through division of the recognized movements of a predecessor [11]. One such visual system imitation learning process is based on a robotic structure proposed by Matarić [12], in which learning is divided into recognition, encoding, and action steps. The recognition step consists of visual tracking and visual attention sub-steps and

involves the extraction of movement data from vision data. In the encoding step, motor primitives to be executed in advance are learned and then classified. Finally, in the motor primitive step, the predecessor is imitated by using selected encoded motor primitives.

Vision-based imitation learning can be expanded into research for generating motor primitives by using spatial-temporal data, in which consecutive sets of data are automatically divided by an SEG-2 algorithm at the maximum point, as calculated by squaring and adding each joint [13]. Another approach for vision system applications involves using an isomap algorithm to reduce the amount of spatial-temporal data [14]. Although conventional isomap algorithms are time-independent and are not normally suitable for such applications, Jenkins's method is time-dependent and is therefore usable here. Jenkins provides another approach to the automatic division of data by isomap algorithms [15] in which the kinematic centroid segmentation (KCS) algorithm is utilized to divide consecutive data. Work has also been undertaken in applying several of the processes described above to intelligent machine architecture (IMA) [16] in which multiple agents communicate with each other.

2.3. Requirements of Intruder Detection Systems

IT security systems are one of the core solutions of Internet of Things (IoT) systems. For IT security systems, important requirements for detecting intruders are required as follows:

- **Convenience**: Given that the configuration of an IT security system is usually very complicated, the setting of the configuration of an IT security system for applying new approaches is more difficult. Therefore, it is necessary to reduce the difficulty of configuration of new approaches when each new approach is applied to intruder detection systems. The configuration includes not only hardware settings but also software settings
- **Generality**: There are a variety of sensors. Therefore, new approaches of intruder detection systems should work without being dependent on specific numbers or types of sensors. In addition, changes in the configuration of sensors should be reflected.
- **Accuracy**: Detecting intruders is the main function of intruder detection systems. Therefore, the accuracy of detecting intruders should be improved and revised when each new approach is applied.

In the proposed method, demonstration-based learning is applied to consider the above requirements. The accuracy of intruder detection systems can be improved by focusing on the data for a specific resident based on demonstration-based learning. The proposed method can be performed on any type of network of sensors. Therefore, it is independent of the sensors. To describe the proposed method well, we utilize the RSSIs of beacon-based tags.

3. Intelligent Intruder Detection Processes

Given that GPS cannot be utilized in indoor environments to detect intruders, beacon RSSIs are utilized in this paper instead of GPS. First, it is necessary to analyze beacon RSSIs to decide whether there are any intruders based on the analyzed RSSIs. This section describes all the processes of the proposed intelligent intruder detection.

3.1. Concept of the Proposed Processes

The proposed processes consist of six phases, including two "living" phases during which residents perform activities of day-to-day living as shown in Figure 1. In the configuring environment phase and deploying sensor phase, the sensors network is configured in cloud computing environments. The sensors network contains diverse kinds of sensors and actuators. The proposed method focuses on RSSIs in the sensors network to recognize notable situations and to detect intruders in cloud computing environments. Devices that invoke RSSIs are not considered in this paper, given that the proposed method can be applied to any devices that invoke RSSIs. The first living phase and learning phase are

performed concurrently to learn normal situations. Then, the second living phase and detecting phase are executed to decide whether or not there is an intruder.

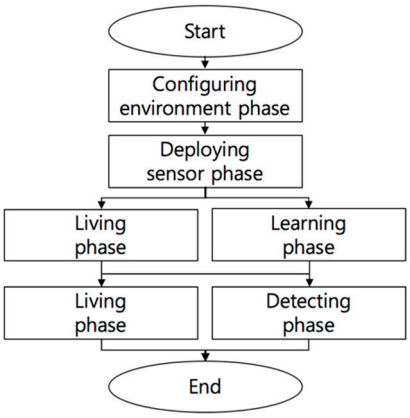


Figure 1. Proposed received signal strenght indicator (RSSI) process phases.

3.2. Learning and Detecting Phases

RSSIs are collected from deployed Access Points (APs) in the measuring stage as shown in Figure 2. The more RSSIs that are gathered, the greater the accuracy of the detection of intruders. RSSI r_i is the ith RSSI, RSSI $r_{i,t}$ is the RSSI r_i at time t, N is the number RSSIs, and the tuple s_t is the ordered set of RSSIs, $(r_{1,t}, r_{2,t}, \ldots, r_{N,t})$, at time t. During the normalization stage, the normalized RSSI $r'_{i,t}$ is calculated using the constant ψ ($r\prime_{i,t} = \frac{r_{i,t}}{\psi} \times \psi$). The tuple s'_t is $(r'_{1,t}, r'_{2,t}, \ldots, r'_{N,t})$. The constant ψ is a positive number that controls the number of types of the tuples, which affects the probability stage. During the filtering stage in the learning phase, the frequency of each tuple is measured as below. The count c_j is the number of the jth tuple of the tuples. The tuples that have low counts usually do not show the constant features. Therefore, the tuples whose counts are smaller than the constant ε are meaningless, and are filtered. The constant ε controls the number of the tuples by removing the meaningless ones. The filtering stage in the detection phase filters tuples by utilizing the counts of the filtering stage in the learning phase.

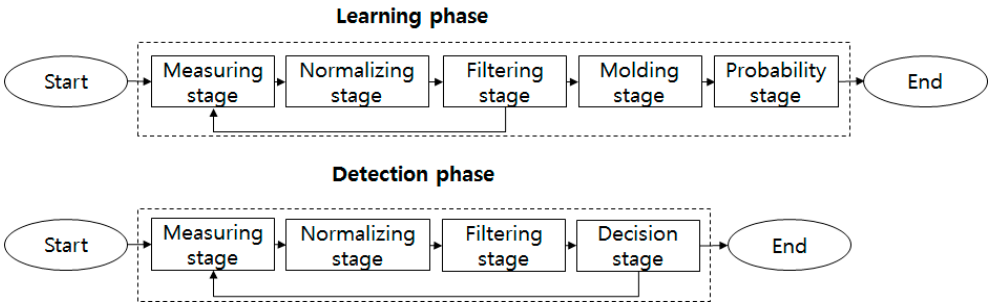


Figure 2. Learning and detection phases.

The molding stage selects valid RSSI ranges. In this paper, an RSSI mold is defined as the boundary of the measurable RSSIs in the tuples. The molding stage consists of four steps: "setting the boundary", "selecting areas", "creating a mold", and "pruning the mold", as shown in Figures 3 and 4. During the "setting the boundary" step, the molding cell size constant δ is determined. All possible RSSI spaces are divided into cells by considering the constant δ. If the constant δ is increased, the number of cells is reduced, and vice versa. Next, during the "selecting area" step, the cells that contain the RSSIs are selected. The "creating a mold" step reduces the size of each cell that contains RSSIs by

considering the maximum and minimum values of the RSSIs within the cell. Finally, the "pruning the mold" step expands all cell sizes by the constant ζ. The constants δ and ζ affect the probability stage.

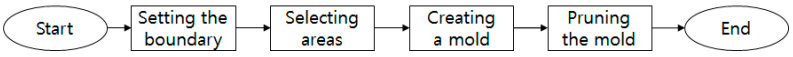
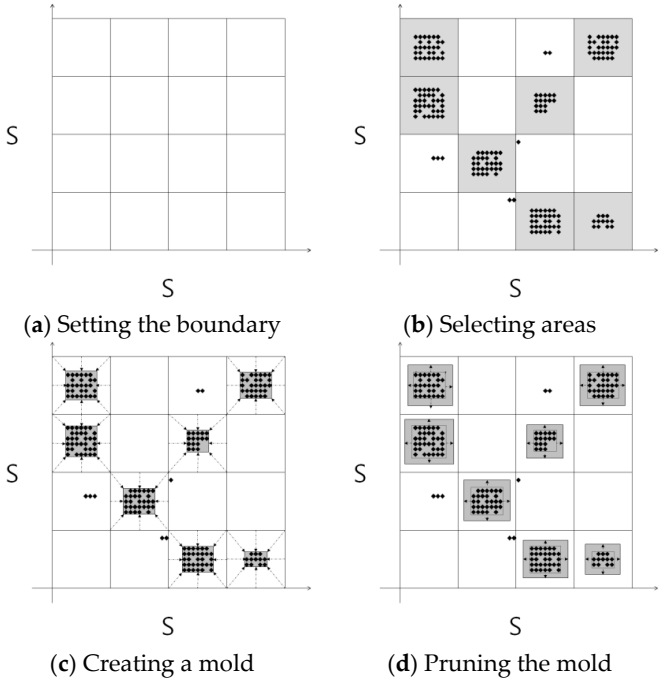


Figure 3. Molding stage.



(**a**) Setting the boundary (**b**) Selecting areas

(**c**) Creating a mold (**d**) Pruning the mold

Figure 4. The concepts of the four steps in the molding stage: (**a**) setting the boundary step; (**b**) selecting area step; (**c**) creating a mold step; (**d**) pruning the mold step.

When RSSIs are measured, it is necessary to analyze their features according to each tag owner. In this paper, Bayesian probability is applied to analyze the features of the RSSIs. The probability stage generates a probability table by considering the top κ counts of tuples as shown in Table 1. The tuple s'_j is the jth tuple when tuples are organized in descending order. The probability $p_{j1,j2}$ is $P(s'_{j2} \mid s'_{j1})$. The above constant ψ controls the probabilities in the probability table. If the constant ψ is increased, the probabilities within each mold become smaller, given that the number of the tuples is increased, and vice versa. The probabilities are also controlled by the constants δ and ζ, given that they control the sizes of RSSI molds.

Table 1. Normalized probability table.

	s'_1	s'_2	...	s'_κ
s'_1	$p_{1,1}$	$p_{1,2}$		$p_{1,\kappa}$
s'_2	$p_{2,1}$	$p_{2,2}$		$p_{2,\kappa}$
...				
s'_κ	$p_{\kappa,1}$	$p_{\kappa,2}$		$p_{\kappa,\kappa}$

The decision stage in the detection phase detects intruders using two approaches. First, intruders are detected by the RSSI molds. The percentages of RSSIs within the RSSI molds for all RSSIs during the detection phase are calculated. If the percentage is less than μ, it is assumed that there is an intruder. Second, the difference between the probability table in the learning phase and the probability table in the detection phase is considered. The difference is the average of all differences between

the probabilities of the probability table in the learning phase and those of the probability table in the detection phase. When the difference is larger than the constant τ, it is assumed that there is an intruder.

4. Experiments

The proposed method can be applied to any kind of sensor network in a cloud computing environment such as in smart buildings and smart houses. In the experiment described here, a laboratory similar to a regular office was configured. This section introduces the configuration of the experimental environments and scenarios, and the proposed method is then validated.

4.1. Experimental Configuration

In the laboratory, there are desks, a printer, a scanner, a washstand, a refrigerator, etc., as shown in Figure 5. To measure the movements of a subject who holds a beacon, four APs are deployed at each corner. Red rectangles denote the APs and blue circles show where the subject usually stays.

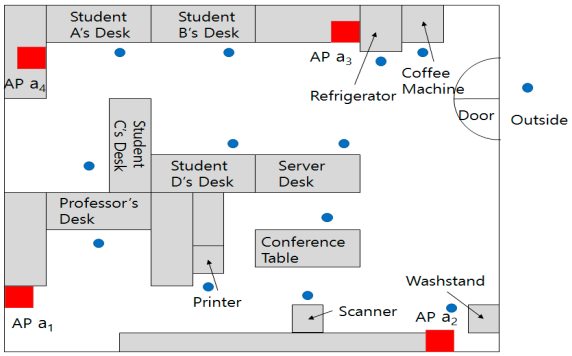


Figure 5. Experimental office environment.

4.2. Sensor Value Analysis

RSSIs of a Bluetooth-based beacon are measured using four APs during the measurement stage in the learning phase. Figure 6 shows the measured RSSIs for one subject in two instances; these are normal RSSIs and are utilizing for learning. Figure 7 shows the results of normalizing the measured RSSIs. The constant ψ was set to 10 considering the number of the normalized RSSIs in each cell to be 8. However, some cells had peak counts compared to other cells. Therefore, when the constant ψ was determined, only the counts of the lower 85% of cells were considered. Table 2 shows the counts of normalized tuples. There are 163 tuples. Tuple $s'_1 = (-85, -85, -75, -65)$ shows the maximum count, 332.

Table 2. The counts of normalized tuples.

AP a_1	AP a_2	AP a_3	AP a_4	Count
−85	−85	−75	−65	332
−85	−75	−75	−65	284
−75	−85	−75	−65	262
−75	−75	−75	−65	210
		...		
−75	−85	−65	−75	5
−75	−75	−65	−85	5
		...		
−95	−65	−65	−85	1
−85	−75	−75	−95	1
−75	−95	−65	−65	1
−95	−95	−85	−55	1

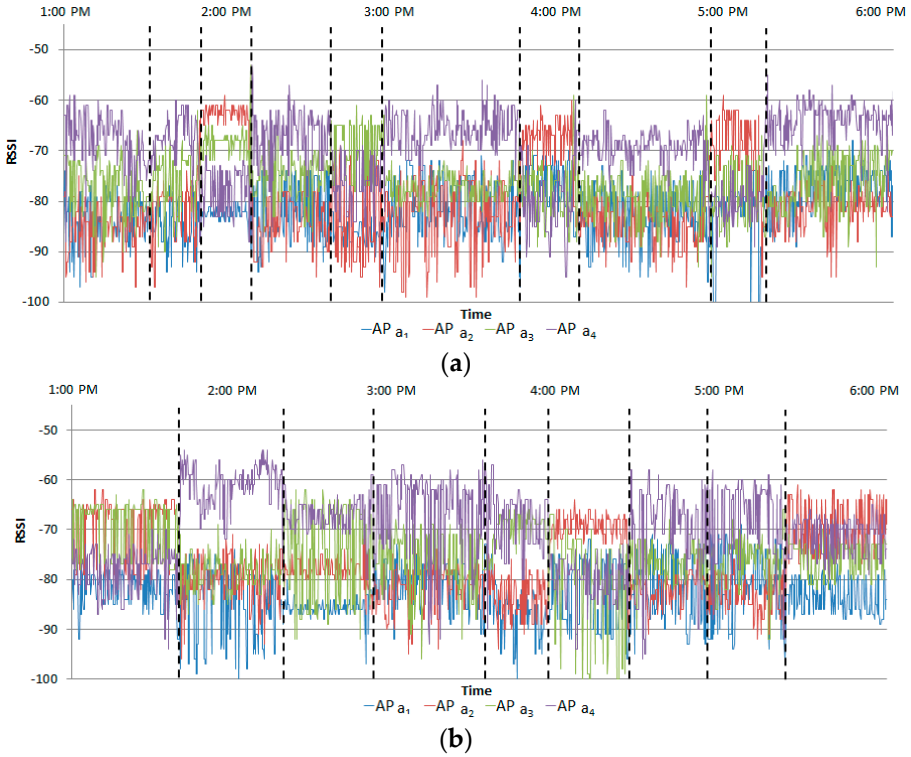


Figure 6. Measured RSSIs for learning. (**a**) The first subject's RSSIs; (**b**) The second subject's RSSIs.

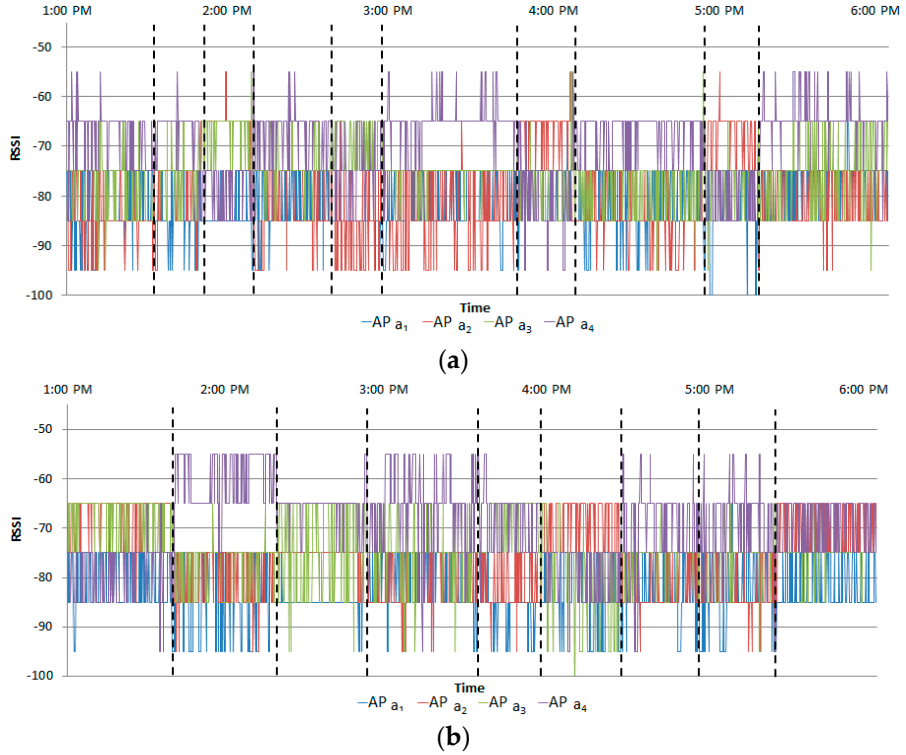


Figure 7. Normalization results. (**a**) The measured RSSIs on the 1st day for a subject; (**b**) The measured RSSIs on the 2nd day for a subject.

Figure 8 shows the counts of normalized tuples as a graph. Given that the constant ε was set as five, 89 tuples were selected and 74 tuples, for which counts were less than five, were filtered. If the

counts are low, the corresponding probabilities become inaccurate. Therefore, the counts less than five were removed in this experiment.

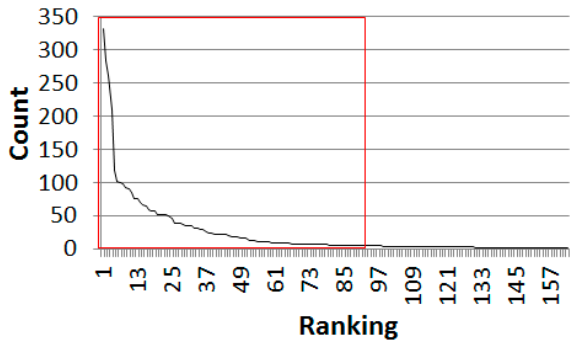
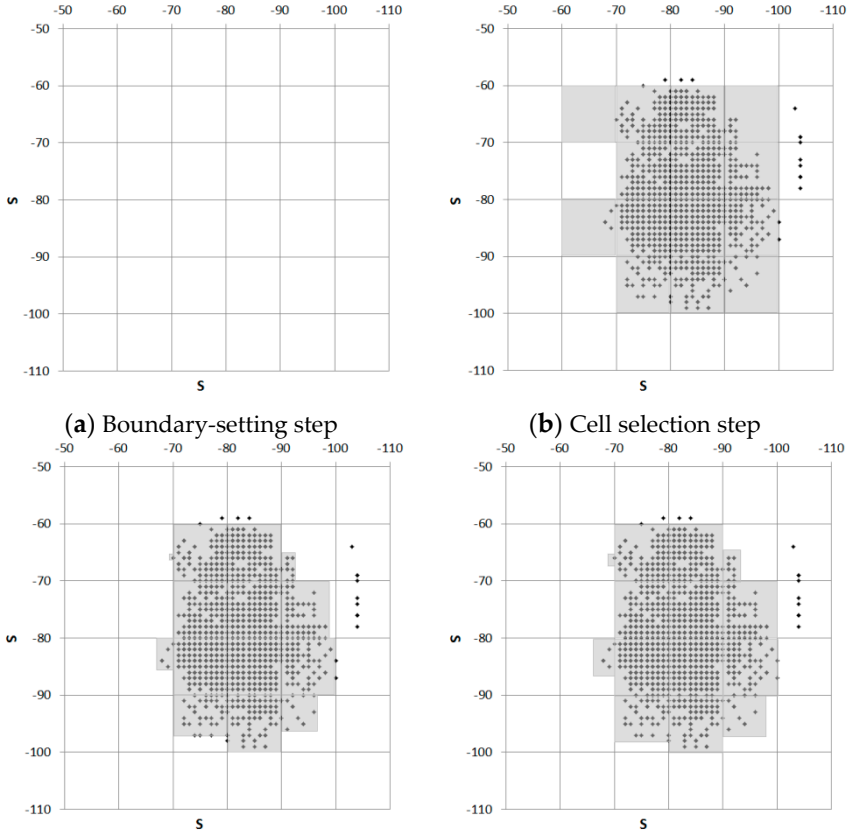


Figure 8. Ordered counts of normalized tuples.

The molding stage was processed utilizing 89 tuples. The measured RSSIs were in the range from −110 to −50. In the experiment, the molding cell size constant δ was set as 10, given that the constant ψ was 10. The molding cell size constant δ could be set as the half of 10. However, if the molding cell size constant δ is set as 10, the counts and probabilities of tables became low and inaccurate. Figure 9b shows the selected cells. Figure 9c shows the generated RSSI molds. In Figure 9d, the RSSI molds are expanded by ζ = 1, considering the molding cell size constant δ, given that the size of the constant ζ was 10% of the size of the molding cell size constant δ.



(**a**) Boundary-setting step (**b**) Cell selection step

(**c**) Mold-creation step (**d**) Mold-pruning step

Figure 9. The results of four steps in the molding stage: (**a**) boundary-setting step; (**b**) cell selection step; (**c**) mold-creation step; (**d**) mold-pruning step.

The probability table in the learning phase is shown in Table 3 below. In the experiment, the constant κ was 10 according to experiments. Given that the subject usually stays in the same place, the probabilities where $j1$ was equal to $j2$ show high counts. Table 4 shows the normalized probability table in the learning phase.

Table 3. Counts of tuples of RSSIs in Learning phase.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	158	28	39	7	1	0	1	13	0	0
s'_2	25	122	8	30	14	0	18	1	0	0
s'_3	42	4	115	24	0	0	0	4	0	0
s'_4	6	24	20	90	1	0	3	0	1	0
s'_5	0	11	2	0	77	0	0	0	0	0
s'_6	0	0	0	0	0	56	2	0	3	11
s'_7	0	11	0	1	0	0	32	10	1	15
s'_8	15	1	5	0	0	0	2	34	0	0
s'_9	0	0	0	2	0	0	0	0	42	7
s'_{10}	0	3	0	0	0	8	4	0	8	41

Table 4. Probability table in Learning phase.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	0.64	0.11	0.16	0.03	0.00	0.00	0.00	0.05	0.00	0.00
s'_2	0.11	0.56	0.04	0.14	0.06	0.00	0.08	0.00	0.00	0.00
s'_3	0.22	0.02	0.61	0.13	0.00	0.00	0.00	0.02	0.00	0.00
s'_4	0.04	0.17	0.14	0.62	0.01	0.00	0.02	0.00	0.01	0.00
s'_5	0.00	0.12	0.02	0.00	0.86	0.00	0.00	0.00	0.00	0.00
s'_6	0.00	0.00	0.00	0.00	0.00	0.78	0.03	0.00	0.04	0.15
s'_7	0.00	0.16	0.00	0.01	0.00	0.00	0.46	0.14	0.01	0.21
s'_8	0.26	0.02	0.09	0.00	0.00	0.00	0.04	0.60	0.00	0.00
s'_9	0.00	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.82	0.14
s'_{10}	0.00	0.05	0.00	0.00	0.00	0.13	0.06	0.00	0.13	0.64

4.3. Experimental Validation Scenarios

To validate the proposed method, experimental scenarios were defined as follows. The first is the scenario when Student A acts in an ordinary way. This scenario is outlined in Figure 10.

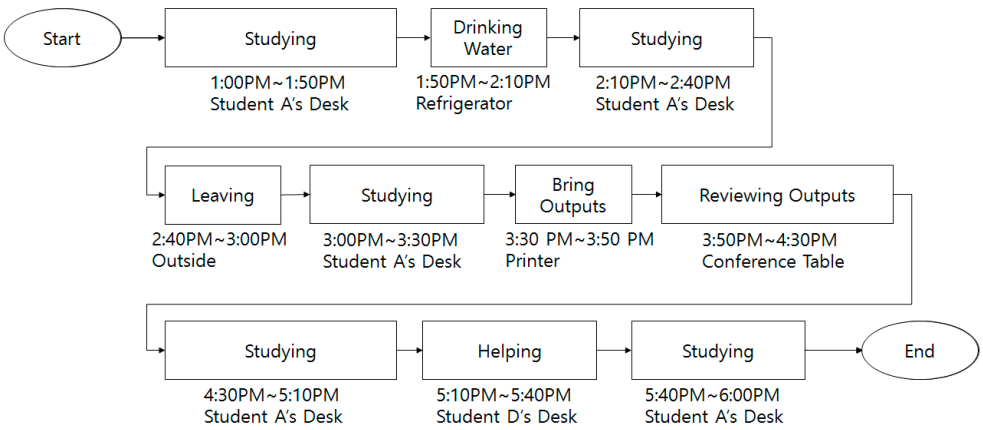


Figure 10. One normal scenario used for validation when a student A acts normally.

Student A performs research from 13:00 to 18:00 in the laboratory. One day, Student A arrives to the school and works on research at his/her desk from 13:00. While working, Student A gets thirsty

and goes to the refrigerator, and then drinks water. After drinking water, Student A returns to his/her desk and works on research. Student A leaves the laboratory to visit the restroom. After returning from the restroom, Student A returns to his/her desk to work. Student A prints the data required for a meeting and goes to the printer to collect the print outs. Student A goes to the conference table with the print outs and sits down. Student A checks the print outs and go to his/her desk. While working on research, Student D asks a question about the research assignment, so Student A approaches Student D to answer the question. After answering, Student A returns to his/her desk to work on research. In this manner, an ordinary scenario is defined and data is collected.

Abnormal scenarios are also defined to simulate the proposed method as shown in Figure 11. Student A goes to the professor's desk and looks for documents to address the work the professor has requested. After finding the documents, Student A returns to his/her desk, calls the professor, and informs the professor of the contents of the document. Student A sits at his/her desk to conduct research. Student A leaves the laboratory to go to another laboratory.

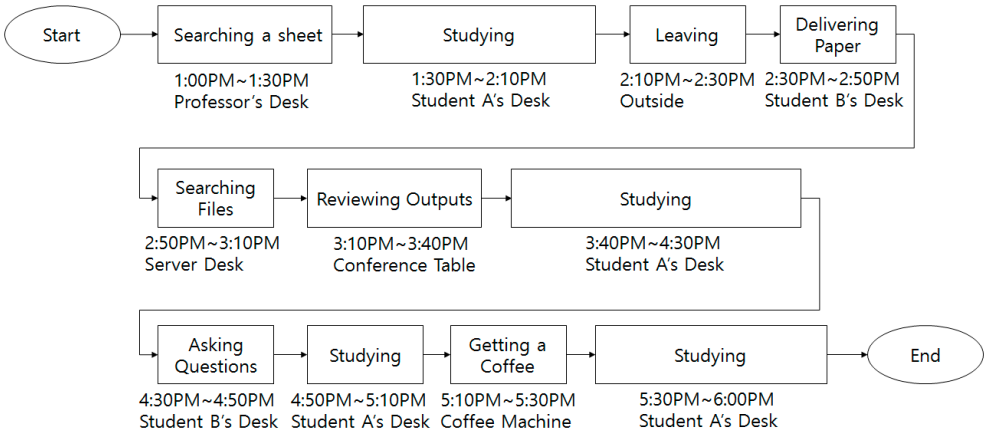


Figure 11. An abnormal scenario for validation when student A moves differently.

On the way back to the first laboratory, student A receives documents to pass on to Student B from another student. Student A enters the laboratory and delivers the documents to Student B. After describing the content of the documents to Student B, Student A goes to the server desk. Student A finds important information saved on the server. Student A sends it through email. Student A goes to the conference table to attend a meeting. After the meeting, Student A goes to his/her desk to work on research. Student A takes questions about the documents he/she received from Student B. Then Student A returns to his/her desk to work on research. Student A becomes tired, so goes to the coffee machine, drinks coffee, and returns to his/her desk to work on research. In this scenario, Student A's behavior is not ordinary because student A goes to the professor's desk and goes to the server desk to access the server computer.

4.4. Sensor Value Measurement

Figure 12 shows the measured RSSIs when the normal scenario was performed by Student A. Figure 13 shows the measured RSSIs when the abnormal scenario was performed by Student A. The RSSI of delivering a paper was similar to that of searching files.

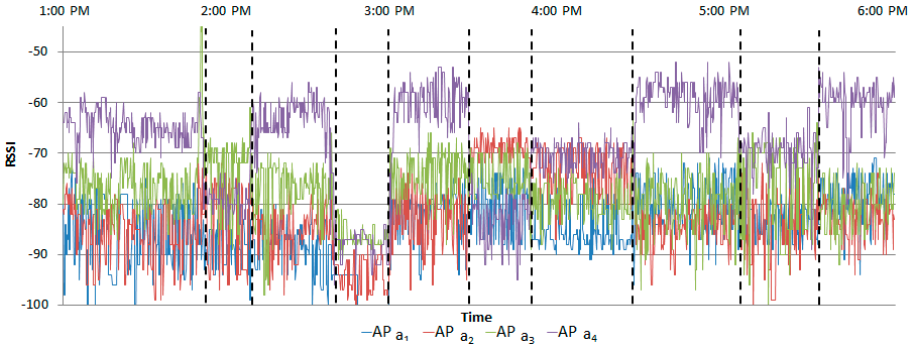


Figure 12. The RSSI of a beacon when Student A acts normally.

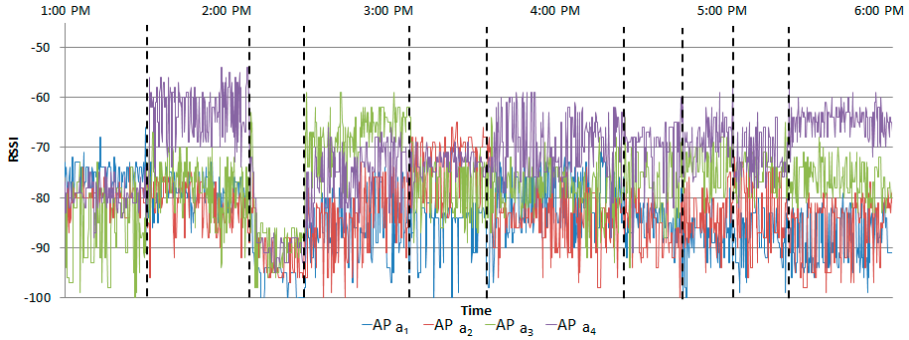


Figure 13. The RSSI of a beacon when Student A moves differently.

Figures 14 and 15 show the normalized RSSIs. The ranges where there are no normalized RSSIs are invoked because of the RSSIs removed by the filtering stage. Figure 16 shows the results of the mold of the normalized and filtered RSSIs.

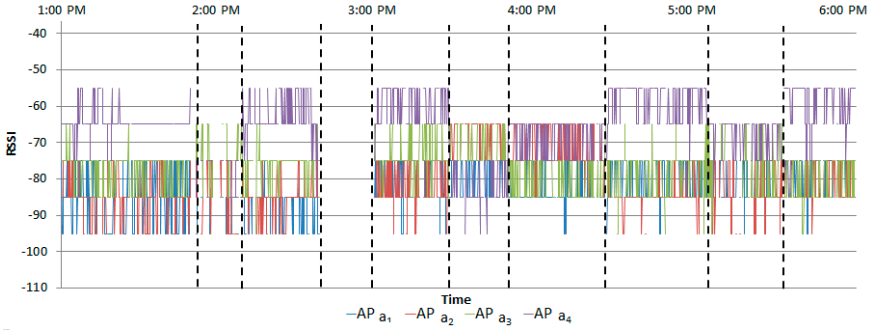


Figure 14. Normalized and filtered normal RSSIs.

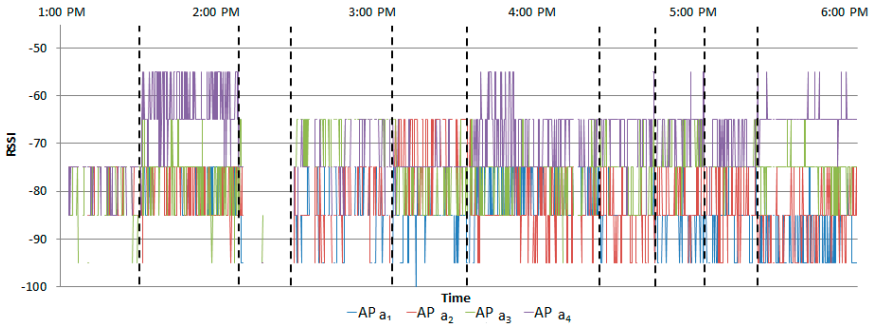


Figure 15. Normalized and filtered abnormal RSSIs.

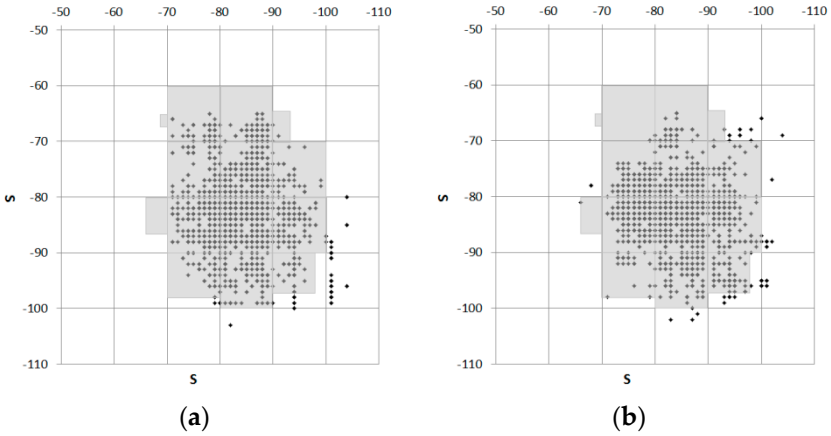


Figure 16. Mold results: (**a**) normalized and filtered normal RSSIs; (**b**) normalized and filtered abnormal RSSIs.

Before comparing probability tables, whether the percentages of normalized and filtered RSSIs are within the RSSI mold for all RSSIs is checked during the detection phase. If the percentage is less than μ, it is assumed that there is an intruder. Figure 17 shows the result of the check for whether the percentages of normalized and filtered normal RSSIs are within the RSSI mold. When the normalized and filtered normal RSSIs are within the RSSI mold, a status of "Correct" is indicated. Other cases are denoted by "Incorrect." The percentage of normalized and filtered normal RSSIs was 80%. As shown in Figure 18, the percentage of normalized and filtered abnormal RSSIs was 79%. Therefore, the two percentages were similar, and larger than the constant μ = 70. The constant μ was set according to experiments.

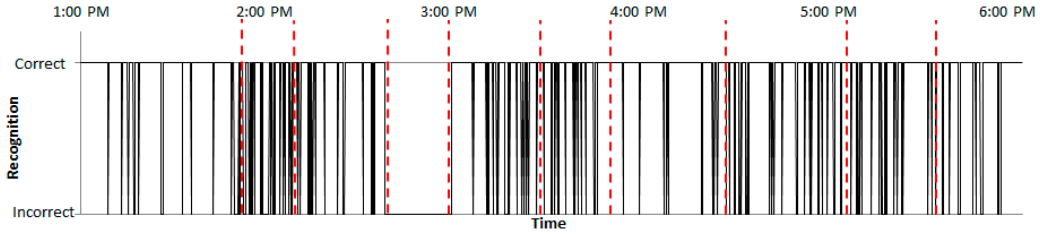


Figure 17. Normalized and filtered RSSIs in normal scenarios compared to RSSIs in the learning phase.

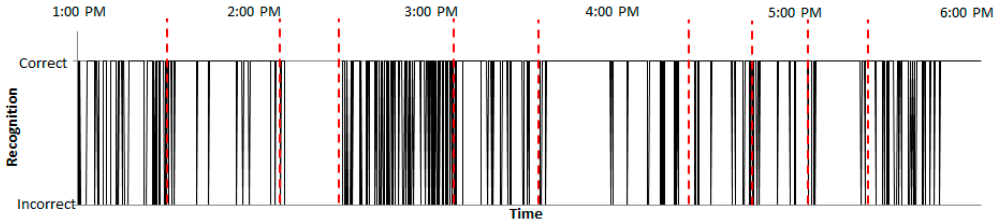


Figure 18. The normalized and filtered abnormal RSSIs in abnormal scenarios comparing to the RSSIs in the learning phase.

Given that the two percentages were greater than the constant μ, the probability tables were calculated. Table 5 shows the counts of tuples of RSSIs for the normal scenario and Table 6 shows the probability table for tuples of RSSIs for the normal scenario. The probability of a normal scenario in Table 6 was calculated based on the count of tuples of RSSIs for the normal scenario in Table 5. Tables 7 and 8 show the results for the abnormal scenario.

Table 5. Counts of tuples of RSSIs for the normal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	63	8	15	0	1	0	0	2	0	1
s'_2	12	44	2	5	3	0	5	0	0	2
s'_3	15	2	23	5	0	0	0	0	0	0
s'_4	2	4	1	13	0	0	0	0	0	0
s'_5	0	2	0	1	4	0	0	0	0	0
s'_6	0	0	0	0	0	1	0	0	1	0
s'_7	0	6	0	1	0	0	11	0	0	3
s'_8	0	0	0	0	0	0	0	0	0	0
s'_9	0	0	0	0	0	0	1	0	5	1
s'_{10}	0	1	0	0	0	1	5	0	3	16

Table 6. Probability table of tuples of RSSIs for the normal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	0.70	0.09	0.17	0.00	0.01	0.00	0.00	0.02	0.00	0.01
s'_2	0.16	0.60	0.03	0.07	0.04	0.00	0.07	0.00	0.00	0.03
s'_3	0.33	0.04	0.51	0.11	0.00	0.00	0.00	0.00	0.00	0.00
s'_4	0.10	0.20	0.05	0.65	0.00	0.00	0.00	0.00	0.00	0.00
s'_5	0.00	0.29	0.00	0.14	0.57	0.00	0.00	0.00	0.00	0.00
s'_6	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.50	0.00
s'_7	0.00	0.29	0.00	0.05	0.00	0.00	0.52	0.00	0.00	0.14
s'_8	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
s'_9	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.71	0.14
s'_{10}	0.00	0.04	0.00	0.00	0.00	0.04	0.19	0.00	0.12	0.62

Table 7. Counts of tuples of RSSIs for the abnormal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	78	10	5	0	1	0	2	13	0	0
s'_2	12	26	2	4	1	0	5	0	0	1
s'_3	5	2	16	9	0	0	0	1	0	0
s'_4	1	2	6	13	1	0	0	0	0	0
s'_5	1	0	0	0	10	0	1	0	0	0
s'_6	0	0	0	0	0	1	0	0	0	1
s'_7	2	6	0	1	0	0	16	9	0	3
s'_8	13	1	2	0	0	1	11	33	0	1
s'_9	0	0	0	0	0	0	0	0	0	2
s'_{10}	0	0	0	0	0	0	1	3	1	21

Table 8. Probability table of tuples of RSSIs for the abnormal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	0.72	0.09	0.05	0.00	0.01	0.00	0.02	0.12	0.00	0.00
s'_2	0.24	0.51	0.04	0.08	0.02	0.00	0.10	0.00	0.00	0.02
s'_3	0.15	0.06	0.48	0.27	0.00	0.00	0.00	0.03	0.00	0.00
s'_4	0.04	0.09	0.26	0.57	0.04	0.00	0.00	0.00	0.00	0.00
s'_5	0.08	0.00	0.00	0.00	0.83	0.00	0.08	0.00	0.00	0.00
s'_6	0.00	0.00	0.00	0.00	0.00	0.50	0.00	0.00	0.00	0.50
s'_7	0.05	0.16	0.00	0.03	0.00	0.00	0.43	0.24	0.00	0.08
s'_8	0.21	0.02	0.03	0.00	0.00	0.02	0.18	0.53	0.00	0.02
s'_9	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
s'_{10}	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.12	0.04	0.81

Tables 9 and 10 are the absolute values of the difference between the learned probability and the probability of the RSSIs for the normal scenario and the abnormal scenario, respectively.

For this experiment, the constant τ was set as 6, given that the possible minimum and maximum of the constant τ are 0 and 10, respectively. The difference for the normal scenario was 4.47 and the difference for the abnormal scenario was 7.21. Therefore, we could assume that the subject in the normal scenario is the subject in the learned scenario and that the subject in the abnormal scenario is not the subject in the learned scenario. As a result, the subject in the abnormal scenario is treated as an intruder.

Table 9. Absolute values of the difference between the learned probability and the probability of the RSSIs for the normal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	0.06	0.02	0.01	0.03	0.01	0.00	0.00	0.03	0.00	0.01
s'_2	0.05	0.04	0.01	0.07	0.02	0.00	0.01	0.00	0.00	0.03
s'_3	0.11	0.02	0.10	0.02	0.00	0.00	0.00	0.02	0.00	0.00
s'_4	0.06	0.03	0.09	0.03	0.01	0.00	0.02	0.00	0.01	0.00
s'_5	0.00	0.16	0.02	0.14	0.28	0.00	0.00	0.00	0.00	0.00
s'_6	0.00	0.00	0.00	0.00	0.00	0.28	0.03	0.00	0.46	0.15
s'_7	0.00	0.13	0.00	0.03	0.00	0.00	0.07	0.14	0.01	0.07
s'_8	0.26	0.02	0.09	0.00	0.00	0.00	0.04	0.60	0.00	0.00
s'_9	0.00	0.00	0.00	0.04	0.00	0.00	0.14	0.00	0.11	0.01
s'_{10}	0.00	0.01	0.00	0.00	0.00	0.09	0.13	0.00	0.01	0.03

Table 10. Absolute values of the difference between the learned probability and the probability of the RSSIs for the abnormal scenario.

	s'_1	s'_2	s'_3	s'_4	s'_5	s'_6	s'_7	s'_8	s'_9	s'_{10}
s'_1	0.02	0.00	0.12	0.00	0.00	0.00	0.02	0.10	0.00	0.01
s'_2	0.07	0.09	0.01	0.01	0.02	0.00	0.03	0.00	0.00	0.01
s'_3	0.18	0.02	0.03	0.16	0.00	0.00	0.00	0.03	0.00	0.00
s'_4	0.06	0.11	0.21	0.08	0.04	0.00	0.00	0.00	0.00	0.00
s'_5	0.08	0.29	0.00	0.14	0.26	0.00	0.08	0.00	0.00	0.00
s'_6	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.50	0.50
s'_7	0.05	0.12	0.00	0.02	0.00	0.00	0.09	0.24	0.00	0.06
s'_8	0.21	0.02	0.03	0.00	0.00	0.02	0.18	0.53	0.00	0.02
s'_9	0.00	0.00	0.00	0.00	0.00	0.00	0.14	0.00	0.71	0.86
s'_{10}	0.00	0.04	0.00	0.00	0.00	0.04	0.15	0.12	0.08	0.19

4.5. Performance Analysis Experiment

Intruders are detected by generating and comparing probability tables. Additional experiments were performed to analyze the differences in the probability tables between authorized persons and intruders to validate the robustness of the proposed method where RSSIs were measured within short distances. While an authorized person and an intruder walked through the same corridor, the RSSIs of those were measured only one time for each during the detecting phase and were utilized to generate probability tables. The intruder walked after watching the authorized person in order to emulate their. Tables 11 and 12 show the probability tables of the authorized person and the intruder, respectively. All parameters were set the same as in the previous experiments except for the molding cell size constant δ and the constant τ, which were set repectively as five and three, given that the size of the walking area was smaller than that of the previous experimental area and that the possible minimum and maximum of the constant τ were zero and five, respectively.

Table 11. Normalized probability table for an authorized person.

	s'_1	s'_2	s'_3	s'_4	s'_5
s'_1	0.14	0.14	0	0	0
s'_2	0	0.18	0.18	0	0
s'_3	0	0.25	0.75	0	0
s'_4	0.16	0	0	0.16	0
s'_5	0	0	0	0	1

Table 12. Normalized probability table of for intruder.

	s'_1	s'_2	s'_3	s'_4	s'_5
s'_1	1	0	0	0	0
s'_2	0	0.18	0.18	0	0
s'_3	0	0.25	0.75	0	0
s'_4	0.16	0	0	0.83	0
s'_5	0	0	0	0	1

The probability differences of the authorized person and the intruder were about 2.99 (smaller than the constant τ) and 4.37 (larger than the constant τ). When the two probability tables are compared, that of the intruder is more different than that of the authorized person by about 1.38. According to this result, it appears that the proposed method can detect the RSSI differences between intruders and authorized persons accurately where RSSIs are measured within short distances. Even though the two people walked through the same route of the same corridor, there was a difference in their walking speeds. Although the proposed method does not consider walking speed directly, the walking speed is reflected when the count table is generated. When a person stays in a specific cell, the count of the corresponding cell is increased. Therefore, the count table is different depending on the person.

5. Conclusions

This paper proposed an intruder detection system based on the measured RSSIs of a beacon. By utilizing deployed APs, RSSIs were measured. Based on the measured RSSIs, the movement patterns of a resident were generated and utilized for detecting intruders. In the experiments, RSSIs were collected twice: once for normal RSSIs and once for RSSIs for validation of the proposed method. For validation, two scenarios were defined: one for a normal situation and the other for an abnormal situation. When the RSSIs of both scenarios were compared to the pre-collected RSSIs, the RSSIs of the abnormal scenario were about 61% more different compared with the RSSIs of the normal scenario. Therefore, intruders in buildings can be detected by considering RSSI differences.

By utilizing the proposed method, the patterns of RSSIs of residents can be utilized as the unique keys of the residents, which has the advantage that the owners of tags do not recognize situations where the owners' RSSIs are collected by APs. In the future, an approach for encrypting measured and collected RSSIs of owners will be researched in order to utilize the RSSIs as unique keys.

Acknowledgments: This research was supported by the Keimyung University Research Grant of 2016.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Nam, K.W.; Yoon, K.L.; Bark, J.S.; Yang, W.S. A Feature Extraction Method for Binary Iris Code Construction. In Proceedings of the 2nd International Conference on Information Technology for Application (ICITA), Harbin, China, 8–11 January 2004.
2. Kouamo, S.; Tangha, C. Fingerprint Recognition with Artificial Neural Networks: Application to E-Learning. *J. Intell. Learn. Syst. Appl.* **2016**, *8*, 39–49. [CrossRef]

3. Liu, H.; Darabi, H.; Banerjee, P.; Liu, J. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Trans. Syst. Man Cybern. Part C* **2007**, *37*, 1067–1080. [CrossRef]
4. Choi, E.; Oh, H.; Kim, I. Non-parametric Estimation for Filtering Erroneous Crowd-sourced WiFi Fingerprints. In Proceedings of the Korea Computer Congress 2013, Jeju, Korea, 15–16 November 2013; pp. 1414–1416.
5. Choi, E.; Oh, H.; Kim, I. Simultaneous Localization and WiFi Fingerprint Mapping Based on Particle Filters. *J. Korea Inf. Sci. Soc. Softw. Appl.* **2013**, *40*, 211–219.
6. Choi, E.; Kim, I. Design of a Crowd-Sourced Fingerprint Mapping and Localization System. *Korea Inf. Proc. Soc. Trans. Softw. Data Eng.* **2013**, *2*, 595–602. [CrossRef]
7. Lee, H.C.; Lee, D.M. A Study on Localization System Using 3D Triangulation Algorithm Based on Dynamic Allocation of Beacon Node. *J. Korea Inf. Commun. Soc.* **2011**, *36*, 378–385. [CrossRef]
8. Gorman, B.; Thurau, C.; Bauckhage, C.; Humphrys, M. Bayesian Imitation of Human Behavior in Interactive Computer Games. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR), Hong Kong, China, 20–24 August 2006; Volume l, pp. 1244–1247.
9. Argall, B.D.; Chernova, S.; Veloso, M.; Browning, B. A Survey of Robot Learning from Demonstration. *J. Robot. Auton. Syst.* **2009**, *57*, 469–283. [CrossRef]
10. Calinon, S.; Guenter, F.; Billard, A. On Learning, Representing, and Generalizing a Task in a Humanoid Robot. In Proceedings of the IEEE Transactions on Systems, Man, and Cybernetics (SMC), Montreal, QC, Canada, 7–10 October 2007; Volume 37, pp. 286–298.
11. Jenkins, O.C.; Matarić, M.J.; Weber, S. Primitive-based Movement Classification for Humanoid Imitation. In Proceedings of the 1st IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000), Cambridge, MA, USA, 7–8 September 2000.
12. Matarić, M.J. Sensory-Motor Primitives as a Basis for Imitation: Linking Perception to Action and Biology to Robotics. In *Imitation in Animals and Artifacts*; MIT Press: Cambridge, MA, USA, 2000; pp. 391–422.
13. Fod, A.; Matarić, M.J.; Jenkins, O.C. Automated Derivation of Primitives for Movement Classification. *Auton. Robots* **2012**, *12*, 39–54. [CrossRef]
14. Jenkins, O.C.; Matarić, M.J. Deriving Action and Behavior Primitives from Human Motion. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Lausanne, Switzerland, 30 September–4 October 2002; pp. 2551–2556.
15. Jenkins, O.C.; Matarić, M.J. Automated Derivation of Behavior Vocabularies for Autonomous Humanoid Motion. In Proceedings of the Second International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS), Melbourne, Australia, 14–18 July 2003; pp. 225–232.
16. Erol, D.; Park, J.; Turkay, E.; Kawamura, K.; Jenkins, O.C.; Matarić, M.J. Motion Generation for Humanoid Robots with Automatically Derived Behaviors. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Washington, DC, USA, 5–8 October 2003; Volume 2, pp. 1816–1821.