

Master's Thesis

Title

**Anomaly Detection in Smart Home Network
by Inspecting Exceptional Operations on IoT Devices
using User's Daily-Life Behavior Learning**

Supervisor

Professor Masayuki Murata

Author

Masaaki Yamauchi

February 8th, 2019

Department of Information Networking
Graduate School of Information Science and Technology
Osaka University

Master's Thesis

Anomaly Detection in Smart Home Network
by Inspecting Exceptional Operations on IoT Devices
using User's Daily-Life Behavior Learning

Masaaki Yamauchi

Abstract

Many devices, such as air conditioners, heaters, and refrigerators, are now being connected to the Internet. These devices are called "IoT" devices. As the number of these IoT devices increases, they have become targets of cyberattacks. Especially, the operations by attackers can cause serious problems, which may harm users or make users unsafe. However, such attacks are difficult to detect using security software or IDS, because the attackers use the same protocol as legitimate operations by users.

In this thesis, we propose a method to detect such attacks based on user behavior. We model user behavior as a sequence of user's events, which includes the operation of IoT devices and other behavior monitored by any sensors such as user's entering and leaving. Our method learns the sequences of events for each one of a set of conditions predefined by the time and sensor values such as room temperature and humidity. After learning, our method detects attacks by comparing the sequences of the events including the current operation with the learned sequences.

For the evaluation of our method, we constructed a network of home IoT devices in our laboratory. We selected four subject students, let them use the devices as they like, and captured all packets of the home network. We obtained times of users' entering/leaving and operating the home IoT devices, using the captured packets, and evaluated our method using the obtained data. When the method observed sequences of events related to the operations, our method detected around 95–100% of anomalous operations with less than 23% of misdetection ratio. Based on the results, we demonstrate the accuracy of our method and discuss the limitations of our method.

Keywords

Anomaly Detection

IoT

Security

Smart Home

Behavior Pattern

Operation by Attackers

Contents

1	Introduction	7
2	Anomaly Detection Based on User Behavior	9
2.1	Learning Model	10
2.1.1	Condition	10
2.1.2	User Behavior	10
2.2	Learning User Behavior	11
2.2.1	Generation of Event Sequences	11
2.2.2	Selection of Conditions	12
2.2.3	Updating the Tree of Event Sequences	12
2.3	Detection	12
2.3.1	Generation of Event Sequences	12
2.3.2	Decision-Making Method	13
2.4	Parameter Setting	14
2.4.1	Setting the Parameter T	14
2.4.2	Setting the Parameters α_i and n_d	16
3	Evaluation	21
3.1	Evaluation Environment	21
3.2	Dataset	24
3.3	Procedure	24
3.3.1	Misdetection Ratio	24
3.3.2	Detection Ratio	25
3.4	Hyperparameters Setting	25
3.5	Evaluation Results	26
3.5.1	Evaluation Results for One Month	26
3.5.2	Evaluation Results for Three Months	28
3.6	Discussion	30
4	Conclusion and Future Work	31

Acknowledgments	32
References	34

List of Figures

1	Overview of Anomaly Detection for Smart Home Based on User Behavior . . .	9
2	Overview of the Detection Method	10
3	Generated Event Sequences when Events A, B, and C Occur	11
4	Experimental Home Network Environment	21

List of Tables

1	List of Deployed IoT Devices	22
2	List of Deployed IoT Sensors	23
3	Hyperparameters for Evaluation	25
4	Detection Results for January 2017	26
5	Detection Results for April, June, and August 2017.	28
6	Detection Results for May, July, and September 2017.	28

1 Introduction

Recently, home appliances and healthcare devices, such as refrigerators, air conditioners, and pacemakers, have started to be connected to the Internet in addition to personal computers and smartphones. These devices are called “IoT (Internet of Things)” devices. Users can obtain information from the IoT devices and can operate the IoT devices with using a smartphone, a tablet, or an AI speaker, such as Google Home [1] or Amazon Echo [2].

Currently, seven billion IoT devices are connected to the network. 215 billion IoT devices will be connected to the network in 2025 [3]. As the number of devices connected to the Internet increases, the risk that these devices become the target of cyberattacks is increasing [4–7] and, in fact, direct attacks and malware targeting IoT devices [8, 9] have already been observed.

Most of the current attacks targeting IoT devices are designed to compromise IoT devices to create botnets [10, 11]. Such attacks may be detectable by methods based on an analysis of the behavior of the attackers [12–14] or in comparison with legitimate usage [15, 16].

However, because IoT devices are closely related to everyday life, there is a risk of attacks having an immediate and personal effect on users [17]. In particular, the operation of IoT devices by attackers may make the users unsafe, and may even harm them, by changing the set temperature of an air conditioner, the setting of a healthcare device, or similar. In addition, in a case that high wattage IoT devices simultaneously operated, power demand would change greatly and the change may cause a large scale blackout [18]. Therefore, methods to detect and prevent operations initiated by attackers are necessary.

Conventionally, security software and intrusion detection systems are used to detect cyberattacks [19]. They detect attacks by pattern matching, comparing the packets with predefined rules or detecting outliers of statistical information of observed traffic. However, because packets sent by attackers to operate IoT devices are the same as the packets sent by legitimate users and packets of operating IoT devices are small, existing methods have difficulty in detecting anomalous operations. In particular, if an attacker sends packets via the compromised smartphone of a legitimate user, the packets cannot be distinguished

from packets sent by legitimate users.

In this thesis, we propose a method to detect the anomalous operation of home IoT devices based on learning users' behavior related to the operation of IoT devices. In this method to detect the anomalous operation, a home gateway that can monitor all packets between the home network and the Internet learns users' behaviors and detect anomalous operations. A home gateway that connects to the home IoT devices, home IoT sensors and smartphones can collect environment information in home obtained from sensors such as room temperature and humidity. Also, a home gateway can get information whether a user is at home based on the information, using information that whether a user's smartphone is connected to the home network [20]. In addition, a home gateway can grasp operations of home IoT devices from the home and via the network, because the home gateway relays packets that operate the home IoT devices via the Internet and from home smartphone. Using such information, a home gateway learns the behaviors of users for each condition defined by the time of day and the information observed by the sensors in the home network. Then, when a command arrives, the gateway checks whether it matches the learned behavior. If the command does not match the learned behavior, the gateway classifies the command as an anomaly.

The rest of this thesis is organized as follows. We describe our method to detect anomalous operations in section 2. Then, we evaluate our method in section 3. Finally, we conclude this work and discuss future issues in section 4.

2 Anomaly Detection Based on User Behavior

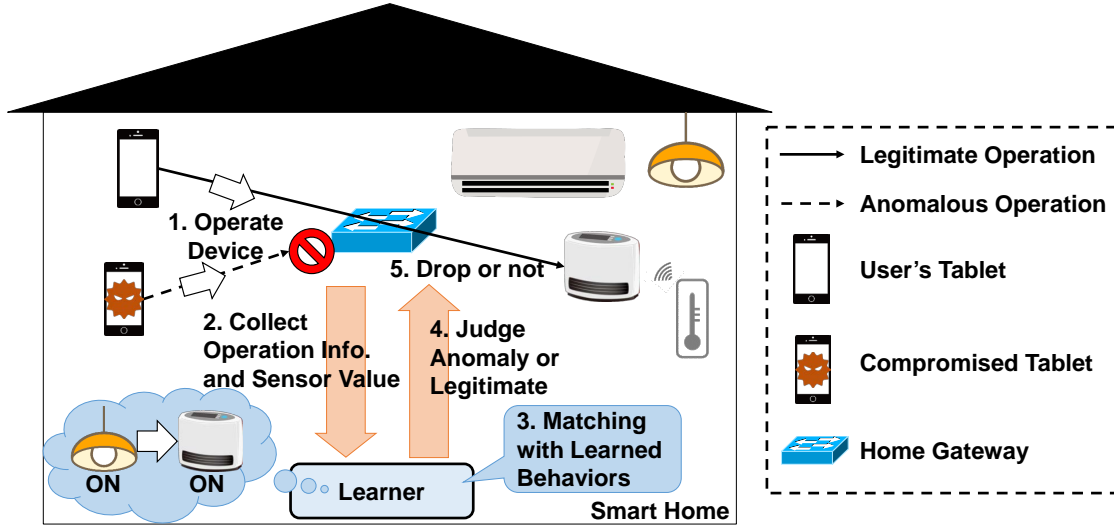


Figure 1: Overview of Anomaly Detection for Smart Home Based on User Behavior

We propose a method to detect the anomalous operation of home IoT devices by attackers, shown in Figure 1. It is based on that users have their own patterns of behavior depending on conditions. For example, when a user returns home and feels cold, they turn on a heater first and then turn on a humidifier; on the other hand, when a user feels warm, they never turn on a heater. In addition, the sequence of operations reflects the characteristics of users: one user first turns on a heater and then turns on a humidifier, while another user first turns on a humidifier.

Our method learns such condition-depend operation sequences as behavior patterns and classifies deviations from these learned patterns as anomalous operations.

In this section, we first explain the model of user behavior used in our method. Then, we explain how user behavior is learned and how anomalous operations are detected.

2.1 Learning Model

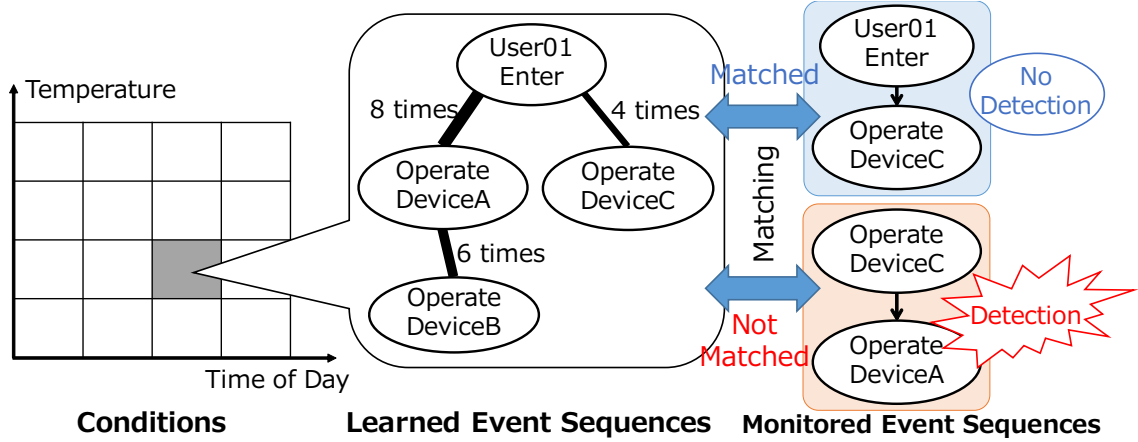


Figure 2: Overview of the Detection Method

Figure 2 shows the model of user behavior in our method.

2.1.1 Condition

We define a condition as a combination of the time of day and values obtained from sensors. The variables representing the various components of a condition will be denoted c_i , where the index i runs from 1 to some maximum value i^{max} . For example, as in Figure 2, c_1 could represent time of day, while c_2 represents the temperature of a room, and so on. Of course, continuous data have to be discretized for tractability. So, we use multiple thresholds for each type of value. A value of c_i that satisfies $c_i^{(j)} \leq c_i \leq c_i^{(j+1)}$ where $c_i^{(j)}$ is the j th threshold for the i th variable, is classified into j th region for that variable.

2.1.2 User Behavior

We define an *event* as any monitored behavior of a user, including the operation of IoT devices and any other behavior monitored by the sensors, such as entering or leaving a room. We define an *event sequence* to be a sequence of events that occurs within T seconds of a previous event. Our method learns a user’s behavior by learning the event sequences for each condition. The set of event sequences corresponding to a condition is modeled as multiple trees whose roots are the first events and whose leaves are the last events. We

detect anomalous event sequences by checking whether the sequence is included in trees. If the sequence is not included in the tree, we classify the sequence as an anomaly.

2.2 Learning User Behavior

Our method learns user behavior from observed information by monitoring events in the home and generating event sequences. Then, the condition corresponding to an event sequence is selected from the predefined set. Finally, the observed sequence is used to update the model for the corresponding condition.

2.2.1 Generation of Event Sequences

Our method regards events observed within T seconds of one another as belonging to the same event sequence, constructing the sequence at the home gateway. However, this means that some event sequences may include events from more than one user because a typical smart home has multiple users. To remove these events, which should be treated as noise, and learn the essential event sequence, we generate multiple event sequences by removing some events from the observed sequence. Figure 3 shows an example of the generation of event sequences. By considering these event sequences, we discover the essential event sequences that include only the events generated by a single user.

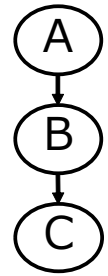
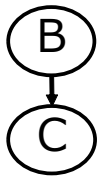
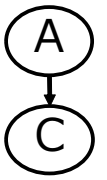
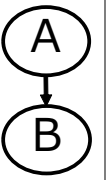



Generated event sequences							
Removed events	None	A	B	C	A,B	A,C	B,C

Figure 3: Generated Event Sequences when Events A, B, and C Occur

2.2.2 Selection of Conditions

Our method selects the condition whose corresponding model is to be updated based on the condition of the first event in a sequence. However, to make effective use of event sequences and learn users' behaviors even when only a small number of event sequences for each condition are observed, we update not only the model corresponding to the condition of its first event but also the models for similar conditions. When the condition of the first event is $\{c_1^{root}, \dots, c_i^{root}\}$, the sequence is used to update the model for the region for which the variable c_i satisfies $c_i^{root} - \alpha_i \leq c_i \leq c_i^{root} + \alpha_i$ for some value α_i , which can be different for each i .

2.2.3 Updating the Tree of Event Sequences

When the home gateway observes an event sequence, we add nodes and links to the tree corresponding to the selected conditions so that the event sequence is included in the tree, the first event of the sequence is the root, and the final event of the sequence is the leaf. Then, we increment a counter for each link on the route corresponding to the event sequence. By repeating this procedure, the count for links related to frequent event sequences increases. Thus, we detect anomalies by using the pruned tree which contains only the links whose counters exceed a threshold $n_d \times L_{num}$ where n_d is a parameter, d is the depth of the link, and L_{num} is the total number of learned operations for the device targeted for detection. By doing so, we eliminate the noise included in the event sequence.

2.3 Detection

When the home gateway observes a new operation, it generates event sequences related to the operation. Then, it compares these sequences with learned behaviors.

2.3.1 Generation of Event Sequences

When the home gateway observes a new operation, it generates multiple event sequences, in the same way as during learning, by removing some events from the sequence of events that occurred within T seconds after previous event and uses these to check whether the operation is anomalous.

2.3.2 Decision-Making Method

We check whether the new operation is anomalous or not by comparing the generated event sequences with the learned behaviors. We first select the condition corresponding to the first operation of the generated sequences. Then, we compare each of the generated sequences with the learned sequences for the selected condition by finding the events in the tree. First, we find the tree whose root is the first event of the event sequence. Then, we find the node that is connected to the found root and is the second event of the event sequence. We repeat searching for nodes until nodes corresponding to all events in the sequence are found or no corresponding nodes are found. We repeat the check for all generated event sequences using the above steps. If all sequences have events whose corresponding nodes are not found in the learned behavior, we classify the operation as an anomaly. On the other hand, if corresponding nodes are found for all events in one of the sequences and the node corresponding to the final event is the leaf, we decide that all the events in the sequence are legitimate. If corresponding nodes are found for all events in one of the sequences but the node corresponding to the final event is not the leaf, we wait for the next event. If the next event does not occur for T seconds, we finish searching for the sequence. If the next event occurs, we generate event sequences including the next event and compare the sequences with learned behaviors using the same steps as described above.

2.4 Parameter Setting

In this section, we describe the method used to set parameters. Our method has three kinds of parameter, T , α_i , and n_d . T affects the generation of event sequences, while α_i and n_d affect the model used for the detection. Thus, we set the parameter T independently from the other parameters so as to generate suitable event sequences. To set parameters, we use events monitored for P weeks. The information on the monitored events includes the kind of event, the time, and the sensor values defining the conditions corresponding to the events.

2.4.1 Setting the Parameter T

First, we set the parameter T . A user may operate multiple devices simultaneously or within a small interval of time. In this thesis, we set T so that the operations of such related devices are included in the event sequences.

Algorithm 1 shows the steps used to set T . In these steps, T is incremented until the generated event sequences become stable. We check whether the generated event sequences are stable by counting the number of pairs of events included in the same event sequence using Algorithm 2. The number of pairs of events included in the same event sequence ($NumOfPair(T)$) initially increases with T . However, the number becomes stable when T is large enough that the operations of related devices are included in the same sequences. Algorithm 2 regards the generated event sequences as stable when the difference between $NumOfPair(T')$ and $NumOfPair(T' + T_{inc})$ is less than N for all T' satisfying $T \leq T' < T + T_{inc}^{stable}$, where T_{inc}^{stable} is a preset parameter.

Algorithm 1 Setting the parameter T

Input: P : Period of the parameter setting (weeks)**Output:** T $T \leftarrow T_{init}$ **while** ISNUMBEROFSEQUENCESSTABLE(T) == *False* **do** $T \leftarrow T + T_{inc}$ **end while****return** T

Algorithm 2 Whether generated event sequences are stable

Input: T **Output:** *True* | *False***function** ISNUMBEROFSEQUENCESSTABLE(T) $T' \leftarrow T$ **while** $T - T' \leq T_{inc}^{stable}$ **do** $T' \leftarrow T' + T_{inc}$ **if** NumOfPair($T' + T_{inc}$) - NumOfPair(T') $\geq N$ **then****return** *False***end if****end while****return** *True***end function**

2.4.2 Setting the Parameters α_i and n_d

After setting the parameter T , we set the parameters α and \mathbf{n} , where $\alpha = \{\alpha_1, \dots, \alpha_{i^{max}}\}$ and $\mathbf{n} = \{n_1, \dots, n_{d^{max}}\}$. i^{max} is the number of values used to define the condition, and d^{max} is a predefined value indicating the depth of the tree considered when setting parameters. For the links whose depth d' is larger than d^{max} , we use $n_{d^{max}}$ as $n_{d'}$.

The parameters α and \mathbf{n} have impacts on the detection ratio (D), that is the proportion of anomalous operations detected as anomalies, and misdetection ratio (M), that is the proportion of legitimate operations detected as anomalies. In this thesis, we set these parameters to obtain the highest detection ratio from among the values that given an acceptable misdetection ratio, $M \leq M^{goal}$. In order to calculate the detection and misdetection ratio, we perform the ‘‘Detection Test’’ shown as Algorithm 4. The detection test calculates the detection and misdetection ratio when anomalous operations of IoT devices are added to event datasets by using cross-validation. At first, we use event datasets to learn behaviors except for a certain day. Then, we add 100 anomalous operations in the omitted day and count the number of detected anomalous operations and misdetected legitimate operations. After doing this for each day, we sum up the numbers of both and calculate the detection and misdetection ratio.

Algorithm 3 shows the steps followed to set parameters α and \mathbf{n} , and we obtain optimal parameters by calling $\text{Opt}\alpha\mathbf{n}(\text{Null}, \text{Null}, k = 1, M^{goal})$. This function finds optimal parameter values by recursively calling itself with k incremented by 1. This function finds optimal parameters by fixing the k' th parameters for $k' < k$. As α_i becomes large, the misdetection ratio becomes small, because the observed event sequences are used to update a larger range of conditions and similar event sequences are regarded as legitimate even if the condition is slightly different from that of the learned sequence. Therefore, if we cannot achieve a misdetection ratio less than M^{goal} by setting $\alpha_i = \alpha'_i$, an acceptable misdetection ratio cannot be achieved for $\alpha_i \leq \alpha'_i$. Similarly, as n_d becomes small, more event sequences are used as legitimate behavior, and the misdetection ratio becomes small. That is, if we cannot achieve a misdetection ratio less than M^{goal} by setting $n_d = n'_d$, an acceptable misdetection ratio cannot be achieved for $n_d \geq n'_d$. Therefore, to reduce the calculation time, we avoid considering such parameters in Algorithm 3.

If no parameters satisfy the target misdetection ratio, our algorithm cannot find suitable parameters. When such cases occurred in our evaluation, we increased the misdetection ratio until suitable parameters could be found.

Algorithm 3 Optimal α , \mathbf{n} , and Detection-Misdetection Ratio

Input: $\alpha^{in}, \mathbf{n}^{in}, k, M^{goal}$ **Output:** $D, M, \alpha^{best}, \mathbf{n}^{best}$

```
function OPT $\alpha\mathbf{n}$ ( $\alpha^{in}, \mathbf{n}^{in}, k, M^{goal}$ )  
  for  $0 < k' < k$  do  
    if  $k' \leq i^{max}$  then  
       $\alpha_{k'} \leftarrow \alpha_{k'}^{in}$   
    else  
       $n_{k'-i^{max}} \leftarrow n_{k'-i^{max}}^{in}$   
    end if  
  end for  
   $(D^{best}, M^{best}) \leftarrow (0.0, 1.0)$   
  if  $k \leq i^{max}$  then  
     $\alpha_k \leftarrow \alpha_k^{max}$   
  else  
     $n_{k-i^{max}} \leftarrow n_{k-i^{max}}^{min}$   
  end if  
  while True do  
    if  $k < i^{max} + d^{max}$  then  
       $(D', M', \alpha', \mathbf{n}') \leftarrow \text{OPT}\alpha\mathbf{n}(\alpha, \mathbf{n}, k + 1, M^{goal})$   
    else  
       $(D', M') \leftarrow \text{DETECTIONTEST}(\alpha, \mathbf{n})$   
       $(\alpha', \mathbf{n}') \leftarrow (\alpha, \mathbf{n})$   
    end if  
    if  $M' > M^{goal}$  then  
      Break  
    else if  $D' > D$  then  
       $(D^{best}, M^{best}, \alpha^{best}, \mathbf{n}^{best}) \leftarrow (D', M', \alpha', \mathbf{n}')$   
    end if  
    if  $k \leq i^{max}$  then
```

```

 $\alpha_k \Leftarrow \alpha_k - \alpha_k^{dec}$ 
if  $\alpha_k < \alpha_k^{min}$  then
    Break
end if
else
 $n_{k-i^{max}} \Leftarrow n_{k-i^{max}} + n_{k-i^{max}}^{inc}$ 
if  $n_{k-i^{max}} > n_{k-i^{max}}^{max}$  then
    Break
end if
end if
end while
return ( $D^{best}, M^{best}, \boldsymbol{\alpha}^{best}, \mathbf{n}^{best}$ )
end function

```

Algorithm 4 Detection Test

Input: α, n

Output: D, M

function DETECTIONTEST(α, n)

for each $Day \in$ Period of parameter setting **do**

 Insert 100 anomaly operations into the Day

 Learn users' behavior without the Day with α

 Detect the Day 's operations with n

end for

 Sum up the results of each Day

return Calculate D, M

end function

3 Evaluation

3.1 Evaluation Environment

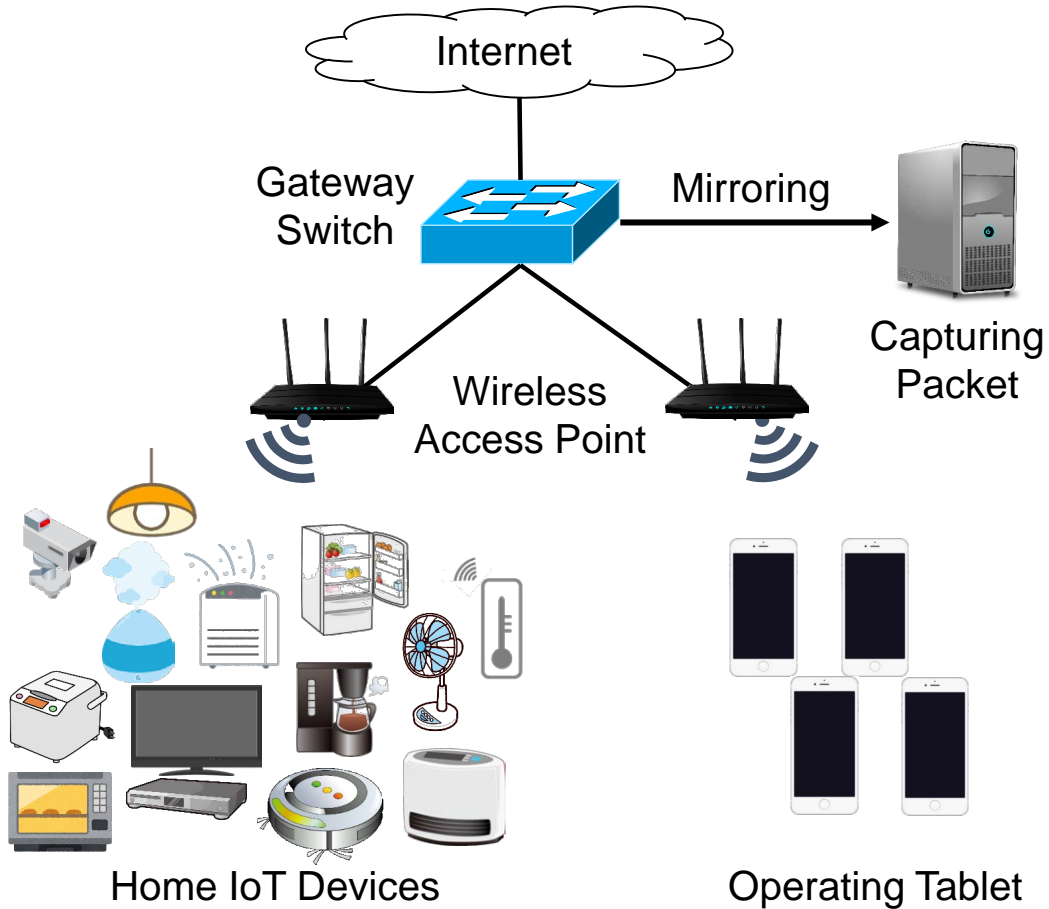


Figure 4: Experimental Home Network Environment

To evaluate our method, we constructed a network of home IoT devices in our laboratory, shown in Figure 4, and captured all packets. Because the “Gateway Switch” in Figure 4 relays all packets of home IoT devices, we can capture every operation packets of home IoT devices. We deployed 13 kinds of connectable home appliances, shown in Table 1 and nine kinds of sensors, shown in Table 2. We selected four subjects of students of the laboratory and let them use the home IoT devices as they liked for a month. In addition, we ask them to record operating logs including time of entering and leaving the laboratory and time of operating home IoT devices. We confirmed the times of opera-

Table 1: List of Deployed IoT Devices

Devices	Number of Set Devices
Air Purifier	2
Automatic Cooker	1
Coffee Maker	1
Electric Fan	4
Heater	1
Humidifier	1
Lighting Equipment	6
Microwave	1
Monitor Camera	1
Recorder	4
Refrigerator	1
Robot Vacuum	1
TV	4

tions of home IoT devices and users' entering/leaving the laboratory can be obtained from captured packets, comparing the time of operations obtained from captured packets with log recorded by subjects. We also captured sensor data. The sensor data includes the temperature, humidity, noise values, and so on.

Table 2: List of Deployed IoT Sensors

Sensors	Number of Set Devices
CO2 Sensor	2
Humidity Sensor	3
Noise Sensor	1
Particulate Matter Sensor	1
Pressure Sensor	1
Rainfall Sensor	1
Temperature Sensor	3
VOC Sensor	1
Wind Sensor	1

3.2 Dataset

We recorded the times of operations of the home IoT devices from captured packets by the increasing or decreasing of the number of packets of each device. From information that whether smartphones are connected to the home network, we generate the events that a user enters or leaves the home. From these information of events and behavior logs recorded by the user, we evaluated our method. However, the value of sensors that includes temperature, humidity, and noise, did not change significantly in our environment. Therefore, in this evaluation, only the time of day is used to define the condition.

3.3 Procedure

We evaluated three datasets; the dataset obtained for one month in January 2017, and the datasets obtained for the three months of April, June, and August 2017 and the three months of May, July, and September 2017. In each period, the same subjects were using the devices. The first dataset was obtained in the winter, and the subjects used the heater. The other datasets were obtained from the spring to the summer, and the subjects did not use the heater but used the electronic fans.

In this section, we use two metrics: the detection ratio and the misdetection ratio.

3.3.1 Misdetection Ratio

To evaluate the misdetection ratio in a straightforward way, we would need a sufficient number of legitimate operations in the test data. However, we have only a limited number of legitimate operations in our collected data. Therefore, we evaluate the misdetection ratio by Leave-One-Out Cross-Validation (LOOCV) [21]. LOOCV separates the dataset into multiple small datasets and verifies one of them after training using the others. After verifying all patterns of the test dataset, we summarize the results.

For this thesis, we separated the data by day and used the data for one day as the test data and used the data for the other days to learn legitimate behaviors.

3.3.2 Detection Ratio

In our evaluation, we added anomalous operations into observed legitimate operations to obtain the detection ratio. We used a strategy similar to that for the evaluation of misdetection to obtain the detection ratio: we separated the dataset by day and used the data for one day for the test and the remaining data for learning.

We added 100 anomalous operations into the test data for each day and tried to detect them. Then, we obtain the detection ratio by counting the total number of detected anomalous operations for all days and dividing by $100 \times \text{number of days of observations}$.

3.4 Hyperparameters Setting

When determining optimal parameter values, we set hyperparameters as Table 3, where P_{num} is the number of operations used for parameter setting for the device targeted for detection.

Table 3: Hyperparameters for Evaluation

Hyperparameter	Value
M^{goal}	0.10
$T_{init}, T_{inc}, T_{inc}^{stable}$	0, 10, 120
N	$10 \times P$
i^{max}	1
$\alpha_1^{min}, \alpha_1^{dec}, \alpha_1^{max}$	600, 600, 43200
d^{max}	2
$n_1^{min}, n_1^{inc}, n_1^{max}$	0.00, $2/P_{num}$, 1.00
$n_2^{min}, n_2^{inc}, n_2^{max}$	0.00, $1/P_{num}$, 1.00

3.5 Evaluation Results

3.5.1 Evaluation Results for One Month

Table 4: Detection Results for January 2017

	Detection Ratio	Detected /Total	Misdetetection Ratio	Misdetetection /Total
Coffee Maker	0.157	346/2200	0.000	0/48
Heater	0.959	2110/2200	0.182	2/11
Humidifier	0.080	176/2200	0.000	0/38
TV A	1.000	2200/2200	1.000	8/8
TV B	1.000	2200/2200	0.000	0/2

First, we used the dataset obtained in January 2017. In this evaluation, we set the parameters by using the data monitored in the first week. After setting parameters, we evaluate our method by LOOCV using the data monitored in the last three weeks. In this evaluation, we also use the data monitored in the first week as training data to learn the legitimate behaviors.

Table 4 shows the results of our evaluation in this case. Over 95% of anomalous operations of the heater were detected. The high detection rate is achieved by using the learned event sequences. Our method learned three patterns of event sequences related to the heater; the patterns that a user operates a humidifier before operating a heater, a user operates a heater before operating a coffee maker, and the user01 leaves the room after operating a heater on the afternoon. Even if an anomalous operation command arrives, we detect anomaly unless the operation matches the above-learned behaviors.

The number of misdetetection of legitimate operations of the heater is two, which is slightly larger than the predefined target ratio M^{goal} . The misdetected operations are related to the event sequence in which a user operates the heater before operating the humidifier. Such event sequences occurred only a few times in one month and our method did not learn such a sequence as legitimate.

Table 4 shows that the detection ratios for operations of the coffee maker or the

humidifier are small. This is caused by *single operation*: for most of the operations of the humidifier or the coffee maker, previous or subsequent events were not observed. In addition, they were operated at various times of day. We set parameters so that such single operations at various times of day cannot be detected as anomalies. As a result, most of such operations are regarded as legitimate.

Table 4 also indicates that all anomalous operations of TVs are detected, but the misdetection ratio for TV A is quite high. This is caused by the low number of operations used to learn legitimate behavior. Furthermore, the monitored event sequences related to TV A vary quite a lot. As a result, an event sequence cannot be learned as legitimate by its similarity to other event sequences. The misdetection ratio of TV B is small because many event sequences are used for detection.

3.5.2 Evaluation Results for Three Months

Table 5: Detection Results for April, June, and August 2017.

	Detection Ratio	Detected /Total	Misdetection Ratio	Misdetection /Total
Coffee Maker	0.611	3908/6400	0.058	3/52
Electric Fan A	0.998	6384/6400	0.000	0/9
Electric Fan B	0.999	6399/6400	0.000	0/6
TV A	0.996	6377/6400	0.171	7/41
TV B	1.000	6400/6400	0.400	4/10
TV C	0.999	6397/6400	0.000	0/10
TV D	0.999	6398/6400	0.111	1/9

Table 6: Detection Results for May, July, and September 2017.

	Detection Ratio	Detected /Total	Misdetection Ratio	Misdetection /Total
Coffee Maker	0.057	368/6500	0.000	0/89
Electric Fan A	0.991	6439/6500	0.074	2/27
Electric Fan B	0.986	6409/6500	0.302	13/43
TV A	0.998	6485/6500	0.231	3/13
TV B	0.999	6497/6500	0.077	1/13
TV C	0.999	6496/6500	0.000	0/3

We also evaluated our method by using datasets for two sets of three months: April, June, and August 2017 and May, July, and September 2017. We used data monitored in the first week of each month to set parameter values and to learn legitimate behaviors, and then performed LOOCV using the remaining data in a similar way and with same parameters as for the 1-month data.

Tables 5 and 6 show the results of the evaluation in these cases. Our method detected more than 98% of all anomalous operations, except for those of the coffee maker. This is

because our method successfully learns legitimate event sequences that are repeated many times, such as the sequence that electronic fans are often turned on immediately after entering the room.

Tables 5 and 6 show that the misdetection ratios for operations of the TVs are smaller than for the dataset for one month. This is because we used more event sequences to learn legitimate behavior. As a result, when a new legitimate operation arrives, the operation can be regarded as legitimate, because similar event sequences have been learned. However, even when we use more data, we cannot detect the anomalous operations of the coffee maker accurately. In addition, the misdetection ratios for TV B in Table 5 and for Electronic Fan B and TV A in Table 6 are also large. This is caused by their single operation. Therefore, to improve the detection of anomalous operations and reduce the misdetection of the legitimate operations, we need a method to accurately identify such single operations as legitimate, which is one of our future research topics.

3.6 Discussion

Our method could detect around 95–100% of anomalous operations if the events related to legitimate operations can be monitored. However, our method also produced several mis-detections. These are either single operations, for which related events are not observed, or rare operations whose corresponding event sequences are rare.

Our method achieves accurate detection of anomalous operations by comparing event sequences. However, we cannot utilize a sequence to check a single operation and the condition of the operation is the only information we have to determine whether such an operation is legitimate.

One approach to improving the accuracy of identification for single operations is to deploy sensors that can monitor events related to these operations. For example, single operations of the humidifier occurred after the water tank became empty. If we can monitor the water tank of the humidifier, we can obtain events related to its operation. Consequently, such operations of the humidifier are no longer single operation. Another approach to improving the accuracy of identification of legitimate single operations is to use more information to define the conditions. In our evaluation, we used only the time of day to define the condition. However, if we define a condition in such a way as to distinguish conditions where legitimate operations are performed from other conditions, we can identify legitimate single operations accurately. Methods to identify the legitimate single operations are included in our future research topics.

The mitigation of misdetection of rare legitimate operations is another challenge, as we cannot obtain a sufficient amount of training data to accurately identify such rare operations in each home. One approach to obtaining more training data is to use data monitored at other homes. However, several issues to be solved before this could be done. One issue is the difference between homes; information about users whose behaviors when using the devices are different may be useless. Thus, we need a method to obtain data from other homes that have users with similar behaviors. Privacy is another issue. That is, we need a method that uses the data from other homes without exchanging private information. This is also one of our future research topics.

4 Conclusion and Future Work

In this thesis, we proposed a method to detect the anomalous operation of home IoT devices. This method learns sequences of user behaviors for each condition that is defined by time of day, sensed temperatures, humidity and so on. Then, when an operation command arrives, the method compares the current sequence with the learned sequences for the condition corresponding to the current condition. If they do not match, our method classifies the operation as an anomaly.

We constructed a network of home IoT devices in our laboratory and let four subjects use the devices. We recorded the times at which the devices were operated along with sensor data. Using this data, we evaluated the detection and misdetection ratios for our method. The results demonstrate that our method detects around 95–100% of anomalous operations if events related to legitimate operations are observed. However, our method cannot accurately identify single operations for which related events are not observed. Rare legitimate operations are also difficult to identify, which results in misdetection. The development of methods to accurately identify rare or single operations is something we will address in future research.

Acknowledgments

This thesis would not accomplish without a lot of great support from many people. First, I would like to express my deepest gratitude to Professor Masayuki Murata of Osaka University, for providing me with the opportunity to research with a talented team of researchers. His creative suggestions, insightful comments, and patient encouragement have been essential for my research activity.

Furthermore, I would like to show my greatest appreciation to Associate Professor Yuichi Ohsita of Osaka University. He devoted a great deal of time for me and gave me a lot of advice about my research. Without his continuous support, my work would not be accomplished. In addition, he enthusiastically taught me the way of thinking and writing. They must be invaluable skills in the future of my life.

Moreover, I would like to show my appreciation to Associate Professor Shin'ichi Arakawa, Assistant Professor Daichi Kominami, Assistant Professor Naomi Kuze, and Specially Appointed Assistant Professor Tatsuya Otoshi of Osaka University for appreciated comments and support. Their kindnesses on my behalf were invaluable, and I am forever in debt.

I am grateful to Dr. Yoshiaki Kato, Dr. Akira Ishihara, Dr. Takeshi Yoneda, Dr. Shotaro Miwa, Dr. Isao Otsuka, Mr. Tsutomu Yoshikawa, Mr. Masao Noguchi, and Mr. Kensuke Ueda of Mitsubishi Electric Corporation for providing me many valuable comments. The opportunity of discussion at the Mitsubishi Electric Cybersecurity Research Alliance Laboratories improved my explanation capability.

I am also grateful to Professor Toru Fujiwara, Assistant Professor Naoto Yanai of Osaka University, and Assistant Professor Yuya Tarutani of Okayama University, for their advice of my research and experimental environment. Their advice from different perspective points of view was always informative and helpful.

I would like to thank subjects who participated in the experiment. Also, I sincerely appreciate a lot of support of Mr. Masahiro Tanaka, Ms. Mari Otokura, and Mr. Takahiro Higuchi of Osaka University for deploying and operating the experimental environment.

I would like to thank all the members of Advanced Network Architecture Research Laboratory at the Graduate School of Information Science and Technology, Osaka University, for their inciting discussions and fellowship.

Last, but not least, I thank my parents and my sister for their invaluable support and constant encouragement during my master studies.

References

- [1] “Google Home,” <https://home.google.com>.
- [2] “Amazon Echo,” <https://www.amazon.com/echo>.
- [3] K. L. Lueth, “State of the IoT 2018: Number of IoT devices now at 7B - Market accelerating,” <https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/>, IoT Analytics, Aug. 2018.
- [4] I. Lee and K. Lee, “The Internet of Things (IoT): Applications, investments, and challenges for enterprises,” *Business Horizons*, vol. 58, no. 4, pp. 431–440, Jul. 2015.
- [5] M. U. Farooq, M. Waseem, A. Khairi, and S. Mazhar, “A critical analysis on the security concerns of internet of things (IoT),” *International Journal of Computer Applications*, vol. 111, no. 7, pp. 1–6, Feb. 2015.
- [6] B. L. R. Stojkoska and K. V. Trivodaliev, “A review of Internet of Things for smart home: Challenges and solutions,” *Journal of Cleaner Production*, vol. 140, no. 3, pp. 1454–1464, Jan. 2017.
- [7] M. Capellupo, J. Liranzo, M. Z. A. Bhuiyan, T. Hayajneh, and G. Wang, “Security and Attack Vector Analysis of IoT Devices,” in *Proceedings of International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, vol. 10658. Springer International Publishing, Dec. 2017, pp. 593–606.
- [8] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, “Understanding the Mirai Botnet,” in *Proceedings of the 26th USENIX Security Symposium*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110.
- [9] D. Palmer, “120,000 IoT cameras vulnerable to new Persirai botnet say researchers,” <https://www.zdnet.com/article/120000-iot-cameras-vulnerable-to-new-persirai-botnet-say-researchers/>, May 2017.

- [10] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, “IoTTPOT: A Novel Honeytrap for Revealing Current IoT Threats,” *Journal of Information Processing*, vol. 24, no. 3, pp. 522–533, May 2016.
- [11] M. Lyu, D. Sherratt, A. Sivanathan, H. H. Gharakheili, A. Radford, and V. Sivaraman, “Quantifying the Reflective DDoS Attack Capability of Household IoT Devices,” in *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, ser. WiSec ’17. New York, NY, USA: ACM, Jul. 2017, pp. 46–51.
- [12] V. Martin, Q. Cao, and T. Benson, “Fending off IoT-hunting attacks at home networks,” in *Proceedings of the 2nd Workshop on Cloud-Assisted Networking*. ACM, Dec. 2017, pp. 67–72.
- [13] K. Xu, F. Wang, and X. Jia, “Secure the Internet, one home at a time,” *Security and Communication Networks*, vol. 9, no. 16, pp. 3821–3832, Jul. 2016.
- [14] S. Shirali-Shahreza and Y. Ganjali, “Protecting Home User Devices with an SDN-Based Firewall,” *IEEE Transactions on Consumer Electronics*, vol. 64, no. 1, pp. 92–100, Feb. 2018.
- [15] K. Xu, F. Wang, R. Egli, A. Fives, R. Howell, and O. McIntyre, “Object-oriented big data security analytics: A case study on home network traffic,” in *Proceedings of International Conference on Wireless Algorithms, Systems, and Applications*, vol. 8491. Springer International Publishing, Jun. 2014, pp. 313–323.
- [16] K. Xu, F. Wang, L. Gu, J. Gao, and Y. Jin, “Characterizing Home Network Traffic: An Inside View,” *Personal Ubiquitous Comput.*, vol. 18, no. 4, pp. 967–975, Apr. 2014.
- [17] N. Komninos, E. Philippou, and A. Pitsillides, “Survey in Smart Grid and Smart Home Security: Issues, Challenges and Countermeasures,” *IEEE Communications Surveys Tutorials*, vol. 16, no. 4, pp. 1933–1954, Apr. 2014.
- [18] S. Soltan, P. Mittal, and H. V. Poor, “BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid,” in *Proceedings of 27th USENIX Security Symposium (USENIX Security 18)*. Baltimore, MD: USENIX Association, Aug. 2018, pp. 15–32.

- [19] B. B. Zarpelao, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, “A survey of intrusion detection in internet of things,” *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, Apr. 2017.
- [20] N. Apthorpe, D. Reisman, and N. Feamster, “A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic,” *arXiv preprint arXiv:1705.06805*, pp. 1–6, May 2017.
- [21] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., Feb. 2006.