Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 210 (2022) 53–60

Procedia Computer Science

www.elsevier.com/locate/procedia

The 13th International Conference on Emerging Ubiquitous Systems and Pervasive Networks (EUSPN 2022)
October 26-28, 2022, Leuven, Belgium

Effective Anomaly Detection in Smart Home by Integrating Event Time Intervals

Chenxu Jiang[a], Chenglong Fu[b], Zhenyu Zhao[a], Xiaojiang Du[c,*]

[a]*Temple University, Philadelphia, PA 19122, USA*
[b]*University of North Carolina at Charlotte, NC 28262, USA*
[c]*Stevens Institute of Technology, Hoboken, NJ 07030, USA*

Abstract

Smart home IoT systems and devices are susceptible to attacks and malfunctions. As a result, users' concerns about their security and safety issues arise along with the prevalence of smart home deployments. In a smart home, various anomalies (such as fire or flooding) could happen due to cyber attacks, device malfunctions, or human mistakes. These concerns motivate researchers to propose various anomaly detection approaches. Existing works on smart home anomaly detection focus on checking the sequence of IoT devices' events but leave out the temporal information of events. This limitation prevents them from detecting anomalies that cause delay rather than missing/injecting events. To fill this gap, in this paper, we propose a novel anomaly detection method that takes the inter-event intervals into consideration. We propose an innovative metric to quantify the temporal similarity between two event sequences. We design a mechanism for learning the temporal patterns of event sequences of common daily activities. Delay-caused anomalies are detected by comparing the sequence with the learned patterns. We collect device events from a real-world testbed for training and testing. The experiment results show that our proposed method achieves accuracies of 93%, 88%, and 89% for three daily activities.

Keywords: Smart Home; Anomaly Detection; Internet of Things

1. Introduction

In recent years, smart home IoT systems have become increasingly popular in the consumer market. In the US, it is reported that smart home IoT devices have entered 43% of households in 2021 [15]. Conveniences brought by smart home platforms, like Alexa and Google Home, also incentive common users to automate their life with smart

* Corresponding author. Tel.: (201) 216-5689.
E-mail address: dxj@ieee.org

1877-0509
10.1016/j.procs.2022.10.119

appliances and sensors. However, security and safety concerns also raise along with the prevalence of smart home IoT devices [6, 9, 17]. Due to limitations on cost and power supply, smart home IoT devices are long known to be unreliable and vulnerable to cyber-attacks, which allows attackers to impose threats to users' safety in the physical world [2, 7]. Besides deliberate attacks, device malfunctions [10] happen even more frequently as some devices are working in harsh environments (e.g., moisture in bathrooms). These device attacks and faults, if not dealt with timely, could cause severe damage. For example, with the help of automation rules [13], an electrical heater could be turned on by low-temperature readings from temperature sensors. False temperature readings that are either caused by sensor malfunctions or attackers' malicious modification could trigger the heater to stay 'on' for a long time, which significantly increases the risk of fire.

To cope with these security issues, there have been many research works [12, 8, 5, 16, 3] that aim to automatically detect the anomalous status of smart home systems. These works model the normal patterns of users' daily activities in the form of invariant event sequences or causal association rules and report deviant patterns as anomalies. For instance, the event sequence "presence detected", "lock unlocked", "front door opened", "hallway motion active", "front door closed", "hallway motion inactive" should be observed when a user goes back to home. Anomaly alarms are raised when the order of this event sequence is violated (e.g., the door gets unlocked without a prior event of presence sensor). However, existing works only focus on the order of events but does not take the events' temporal information into consideration. In some anomalous cases, the sequences of events remain identical to the learned one but with certain events being delayed. In the example case of user going back to home, If an intruder gets the presence sensor and enters the house, the event sequence could also be "presence detected", "lock unlocked", "front door opened", "hallway motion active", "front door closed", "hallway motion inactive", but the time interval between every two events is different. For instance, while the user closes the door immediately after entering the hallway, the intruder may leave the door open until he leaves the house. The difference in the time interval is the key to identifying anomalous cases.

In this paper, we fill this gap by proposing a novel anomaly detection method that takes events' temporal information into consideration. With the additional information, we are able to more accurately profile normal behaviors of smart home IoT systems. More specifically, we enhance the existing event patterns by adding intervals between every two events in the sequence. In addition, we design an innovative scoring mechanism to compare the temporal similarity between two event sequences. Intervals that are either too small or too large will result in low or high scores and trigger anomaly alarms. To evaluate the proposed method, we collect the event sequences of the user's daily activity in a real-world testbed over 10 days. We measure 3 activities that include "Come back home", "Go to work", "Use toilet". The evaluation results show that the accuracy for three activity is 93%, 88%, 89%, respectively.

We make the following contributions:

- We propose a new learning model that can learning time-interval from event sequence. With the help of this new learning model, we can better profile user behaviors.
- We propose a scoring method for event sequence, which gives a score based on the similarity between a test event sequence and the ground truth. The score result is used to decide whether to trigger an anomaly alarm.
- We evaluate the new learning model and scoring method with a real-world testbed and three activities. The result shows that the accuracy for each activity is 93%, 88%, 89%, respectively.

The rest of the paper is organized as follows. We introduce related work in Section 2. In Section 3, we describe the background of the smart home platforms. We introduce the motivation and threat model in Section 4. In Section 5, we introduce the design of anomaly detection system. The evaluation is presented in Section 6. Conclusion is presented in Section 7.

2. Related work

With the rapid development of IoT devices and smart home, the security issue in smart home draws much attention and there has been many papers focusing on the anomaly detection in smart home[12, 5]. Most of them are based on the order of events. For example, AEGIS observes the change in device behavior based on user activities and builds a contextual model to differentiate benign and malicious behavior[12]. Reference [5] precomputes sensor correlation

and the transition probability between sensor states and finds a violation of sensor correlation and transition to detect and identify the faults. Most recent works have applied semantic information extraction [4] to assist the protection of smart home IoT system. For example, authors of PFirewall [3] utilize the semantic information of automation rules to protect smart home users' privacy by filtering out unnecessary events towards cloud servers and HAWatcher [8] mines inter-device correlations and use them to detect device anomalies.

The main difference between these existing anomaly detector and our work is that our paper takes time interval into consideration and propose a novel scoring mechanism method for anomaly detection. To the best of our knowledge, time-interval between every two events isn't used for anomaly detection in the smart homes prior to our work.

3. Background

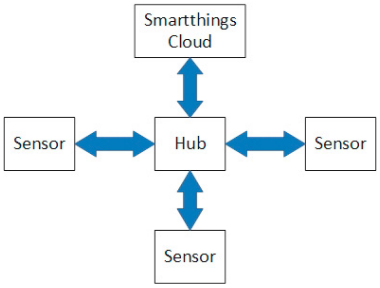

Fig. 1. The architecture of smartthings platform

There are many smart home platforms now, like Smartthings, Amazon Alexa. These smart Home platforms provide so many types of sensor that can be used in smart homes. These sensors have different functions. In this paper, we use smartthings, which is a popular smart platform. The architecture of smartthings is shown in Figure 1. Sensors are connected to each other and smartthings cloud via hub. As a result, sensors can upload its data to smartthings cloud for further analysis and user can control sensor via smarthings cloud. Automation rule is an important part of smart home platform, which is set by user. When the status of the sensor meets the prerequisite of the automation rule, the status of another sensor would change. For example, the automation rule could be "when the door in bathroom open, the light in the bathroom should be turned on/off". As a result, when the user opens the door to walk into bathroom at the first time, the light in the bathroom is turned on. Then when user opens the door to walk out the bathroom, the bedroom light is turned off.

Table 1. Example events collected from smart home IoT system.

Timestamp	Device	Value
10/1/2021 13:00:01	motionSensor	active
10/1/2021 13:00:02	light	on
10/1/2021 13:01:00	light	off
.....	.....	.....

User's activity will active multiple sensors, which would be recorded in the log. The log can be downloaded from smartthings app, which can be simplified as Table 1. When a sensor is activated, the log will record its timestamp, device name, and value at the moment. The value could be numerical values, like "56.0", or a boolean value.

4. Threat Model

In this paper, we consider smart home anomalies that are caused by the following reasons:

- **Faulty devices**. Smart home devices may lose the connection to the hub at any time and have no response to the user's behavior if they are broken up. When the device is offline, there would be no record about this device in the log. For example, when the motion sensor is out of power, it will be offline and lose connection with the hub. So, the smart home system can't detect any motion, and the automation rule based on the motion sensors will be invalid.
- **Large Delays**. When the sensor is activated, it should report the event immediately to the hub, and this event will be added to log in smartthings cloud. But, due to interference from other devices, the sensor may report the event to the hub with a large delay. Large delays would cause disorder in log. Also, the automation rule based on the event will be invalid because the prerequisite of it isn't met, which may lead to a risky consequence. For example, the door may be left unlocked caused by a loss of presence-off event when the user leaves home[8].
- **Intruder**. We use the event sequence in log to profile the user's daily activity. When an intruder enters the building, since its behavior is different from the user, the event sequence pattern changes. For example, when user go back to home, the behavior could be "open the door, close the door, enter the hallway, enter the kitchen". But, when intruder enter the building, its behavior may be "open the door, close the door, enter the hallway, enter the bedroom ". Since the behavior is different, the event sequence will not be the same. Also, even intruder follows the pattern of user's daily behavior, the time interval between each two event may be not the same. For example, user often takes a nap in the bedroom while the intruder just takes a look at the bedroom, the time interval between "open the bedroom door" and "close the bedroom door" would be significantly different.
- **Anomalous activities**. Since the user's behavior is profiled based on previously collected event sequence. Once anomalous activity happens, the event sequence will be different and can be regarded as an anomaly. For example, if the user takes a shower, the event sequence is "open the door, active motion sensor, close the door, active motion sensor, open the door". But, if the user have a fall and gets in shock, the event sequence is "open the door, active motion sensor, close the door," which is different from the previous one.

Due to the reasons mentioned above, the anomalous cases of event sequence is different from normal case of event sequence in two aspects: 1) The order of events is switched; 2) The time intervals between events change.

5. System Design

To cope with the aforementioned anomalies, researchers proposed a series of anomaly detection methods [11] that aim to provide early warnings on anomalous situations before causing damage. Our method improves the existing ones by merging time-intervals between events into the anomaly detection model. In this section, we first introduce our formal representations of event sequence patterns and then discuss how we deal with the additional temporal information using our innovative scoring mechanism.

5.1. Sequence Pattern Representation

We use E to represent device events. For a specific event of a device, we represent it in the form of $E^{state}_{attr}(Device)$. For example, the event of "bedroom motion sensor detect active motion" is represented as $E^{active}_{motion}(bedroom\ motion)$. Users' common daily activities usually trigger multiple devices' events consecutively, which forms certain sequence patterns. For example, if a user uses the toilet, he would open the door, turn on the light and close the door. As a result, the contact sensor in the door, light in the toilet, and motion sensor in the toilet are activated. Aside from the sequence of events, we also take the time interval between every two adjacent events into consideration. In the example above, we also record the interval between "Open the door, active motion sensor, turn on light, close the door". As a result, the event sequence pattern of a user's activity can be represented in (1). $E^{\alpha}_{A_1}(D_1)$ means the event of attribute A_1 on device D_1 turns to state α. While, Δ_i stands for the time interval between the (i)th and the $(i+1)$th events.

$$P(Act_i) = \begin{cases} [E^{\alpha}_{A_1}(D_1), E^{\beta}_{A_2}(D_2), \ldots, E^{\gamma}_{A_n}(D_n)] \\ [\Delta_1, \Delta_2, \ldots, \Delta_{n-1}] \end{cases} \tag{1}$$

5.2. Scoring Mechanism

With the repeat of user's daily activities, there would be many frequent sequences in the log of event sequence. For example, if the user uses the toilet, he would like to open the door, turn on the light and close the door. So after the repeat of this activity, a frequent event sequence "open the door, turn on light, close the door" in the log is observed. After getting the frequent event sequence, we can extract all event sequence which is identical to the frequent event sequence from the log. Then we can get the average time interval of each two events using the extracted event sequence. The frequent event sequence and the vector of average time-interval of each two events can be used to represent the ground truth for this frequent event sequence pattern, which is:

$$P(Act_g) = \begin{cases} [E^{\alpha}_{A_1}(D_1), E^{\beta}_{A_2}(D_2), \ldots, E^{\gamma}_{A_n}(D_n)] \\ [\overline{\Delta_1}, \overline{\Delta_2}, \ldots, \overline{\Delta_{n-1}}] \end{cases} \tag{2}$$

Where $E^{\alpha}_{A_1}(D_1), E^{\beta}_{A_2}(D_2), \ldots, E^{\gamma}_{A_n}(D_n)$ is the frequent event sequence, $\overline{\Delta_i}$ is the average time interval between event $E^{\beta}_{A_i}(D_i)$ and event $E^{\eta}_{A_{i+1}}(D_{i+1})$. Compared with the ground truth of event sequence, the test event sequence is missing some event, or the time interval between every two events is abnormal. Test event sequence can be represented as Equation 3, where $P(Act_t)$ is the test event sequence, k is the number of events, and $k \leq n$.

$$P(Act_t) = \begin{cases} [E^{\alpha}_{A_1}(D_1), E^{\beta}_{A_2}(D_2), \ldots, E^{\gamma}_{A_k}(D_k)] \\ [\overline{\Delta_1}, \overline{\Delta_2}, \ldots, \overline{\Delta_{k-1}}] \end{cases} \tag{3}$$

We propose a scoring mechanism to quantify the similarity between test event sequence and ground truth, which can be represented as in Equation 4, where *score* is the score of test event sequence, num_t is the number of event in test event sequence, num_g is the number of event in ground truth, α is a coefficient, $\theta(ti_t, ti_{g'})$ is the angle between vector ti_t and vector $ti_{g'}$, ti_t is the time-interval vector of test event sequence, $ti_{g'}$ is the modified time-interval vector of ground truth. Since some event is missing in test event sequence, the event doesn't exist in test event sequence is deleted from ti_g to make sure ti_t has the same length with $ti_{g'}$.

$$score = score_n + \alpha * score_c, \quad score_n = \frac{num_t}{num_g}, \quad score_c = 1 - \frac{\theta(ti_t, ti_{g'})}{\pi}, \quad \theta(ti_t, ti_{g'}) \in [0, \pi] \tag{4}$$

6. Performance Evaluation

6.1. Experiment Setup

We evaluate our proposed anomaly detection method on a real-world testbed as shown in Figure 2. All used devices' labels, attributes, and installation locations are listed in Table 2. The testbed consists of two bedrooms, a dining room, and a bathroom and has one resident. In four rooms, we install 14 IoT devices of 4 types. All devices are connected to a SmartThings hub and managed using the SmartThings mobile app [14].

Table 2. IoT devices used in the testbed, their abbreviation labels, attributes, and deployment information

Label	Device Name	Attributes	Deployment
M	SmartThings Motion Sensor	motion	on wall
C	SmartThings Contact Sensor	contact, acceleration	on doors
L	SmartThings Light Bulb	switch	as ceiling light, lamp
V	ThreeReality Smart Switch	switch	to control fan

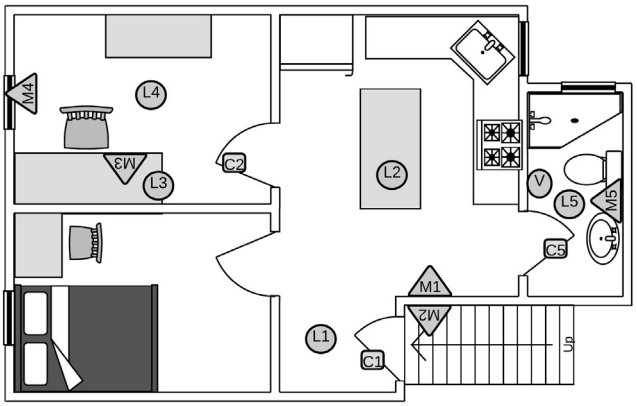

Fig. 2. Floor plans of the testbed and device deployment layout.

Table 3. Activities used to evaluate the proposed method

Activity	Behavior	devices	Automation rule
Come back home	User opens and closes the front door, enters kitchen	M2,C1,L1, M1,L2	$E^{active}_{motion}(M2) \rightarrow E^{on}_{light}(L1)$, $E^{active}_{motion}(M1) \rightarrow E^{on}_{light}(L2)$
Use toilet	User enters the bathroom, closes the door, takes a pee, opens the door and walks out	M5,C5, L5,V	$E^{active}_{motion}(M5) \rightarrow E^{on}_{light}(L5)$, $E^{close}_{contact}(C5) \rightarrow E^{on}_{switch}(V)$, $E^{open>20s}_{contact}(C5) \wedge E^{inactive>20s}_{motion}(M5) \rightarrow E^{off}_{light}(L5) \wedge E^{off}_{switch}(V)$
Go to work	User opens the door, walks into study and sits before the desk	C2,M4, L4, M3,L3	$E^{open}_{contact}(c2) \rightarrow E^{on}_{light}(L4)$, $E^{active}_{motion}(M3) \rightarrow E^{on}_{light}(L3)$

6.2. Data Collection and Augmentation

On this testbed, we evaluate our method on 3 activities that include: "Come back home", "Go to work" and "Use toilet". For each activity as listed in Table 3, we deploy some automation rules according to the participant's requirements and collect about 50 instances for each activity during 10 days experiment. All events are collected from SmartThings mobile app. To improve the effectiveness of the collected data, we augment the collected events of activity instances. More specifically, we use the synthetic minority over-sampling technique (SMOTE) [1] to expand the collected dataset. SMOTE is mainly used in over-sampling the minority (abnormal) class and under-sampling the majority (normal) class to achieve better classifier performance. Because the difference between the nearest neighbor and the selected data would be tiny, we randomly select two instances of normal data rather than select k-nearest neighbors. The detailed process can be represented as below, where P_i and P_j is the vector of the collected data, P_{new} is the vector of the created data.

$$P_{new} = P_i + 0.5 * (P_j - P_i) \tag{5}$$

Since anomalies rarely happen during the experiment period, we generate data of abnormal instances by simulating anomalies based on data of normal instances. We simulate anomaly cases in two methods: 1) deleting one event from a normal event sequence; (2) extending a one-time interval in an event sequence by 50 times. We call these two types of anomaly cases as "Anomaly (seq)" and "Anomaly (ti)", respectively.

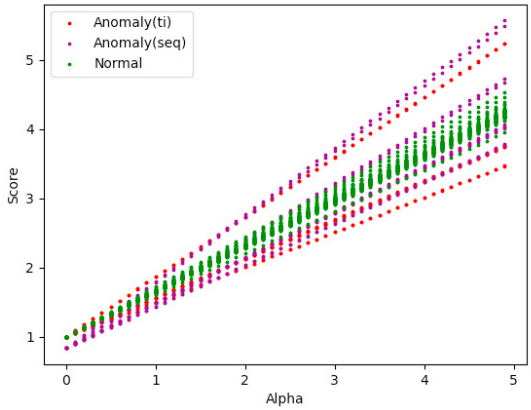

Fig. 3. The influence of α on the scores. Green, red, and purple points are for normal, anomaly (seq), and anomaly (ti) instances, respectively.

6.3. Training Process

In the training process, we use collected normal and anomaly instances to find the optimal α and score section. We use 40 instances of the normal data, 10 instances of anomaly (seq) data, and 10 instances of anomaly (ti) for each activity. Equation (4) shows that we can calculate the score of the test event sequence by comparing it with a given reference sequence. The score is affected by the value of α. To explore the impact of the value of α, we calculate the score of all data instances in the testing dataset with different α values in the range of $[0, 5]$. From Figure 3, we can select the α that the cluster of the green points is obviously deviated from the cluster of purple point and red point. For each selected α, we can find the score section in which the green point dominates over the purple point and red point. So this score section can be selected as the threshold to distinguish normal data instances from anomaly instances. By setting α as 3, we can get the score threshold boundaries of $[2.9, 3.1]$, $[3.45, 4]$, and $[2.5, 2.62]$ for normal activities of "Come back home", "Go to work", and "Use toilet", respectively.

Table 4. Testing results of three activities.

Activity	Case	Amount	TP+TN	FN+FP	Accuracy
Go back home	anomaly (seq)	20	20	0	100%
	anomaly (ti)	20	16	4	80%
	Normal	60	57	3	95%
	Total	**100**	**93**	**7**	**93%**
Go to work	anomaly (seq)	20	9	11	45%
	anomaly (ti)	20	20	0	100%
	Normal	60	59	1	95%
	Total	**100**	**88**	**12**	**88%**
Use toilet	anomaly (seq)	20	20	0	100%
	anomaly (ti)	20	9	11	45%
	Normal	60	60	1	100%
	Total	**100**	**89**	**11**	**89%**

6.4. Testing Results

In the testing process, we apply the learned score range in the training process to 100 testing data instances for each activity to check the performance of anomaly detection. Each activity contains 20 instances of anomaly (seq) data, 20

instances of anomaly (ti), and 60 instances of normal data. None of the testing data is used in the training process. We evaluate the performance of our work by using accuracy as the metric, which is calculated by following the Equation 6. As shown in Table 4, our work achieves the accuracy of 93%, 80%, and 89% in terms of three activities, respectively.

$$Accuracy = \frac{True\ Positive\ +\ True\ Negative}{True\ Positive\ +\ True\ Negative\ +\ False\ Positive\ +\ False\ Negative} \tag{6}$$

7. Conclusion

In this paper, we proposed a new learning model that takes time intervals among events into consideration. We designed a new score metric to quantify the temporal similarity between a testing event sequence and a reference sequence of ground truth. We evaluated the performance of the proposed method with 10 days experiment in a real-world testbed. The results showed our method achieves accuracies of 93%, 88%, and 89% for three testing activities.

Acknowledgements

This work was supported in part by the United States National Science Foundation (NSF) under grants CNS-1828363, CNS-2204785, and CNS-2205868.

References

[1] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research 16, 321–357.
[2] Chi, H., Fu, C., Zeng, Q., Du, X., 2022. Delay wreaks havoc on your smart home: Delay-based: Automation interference attacks, in: 2022 IEEE Symposium on Security and Privacy (S&P), IEEE Computer Society. pp. 1575–1575.
[3] Chi, H., Zeng, Q., Du, X., Luo, L., 2021. Pfirewall: Semantics-aware customizable data flow control for smart home privacy protection, in: Network and Distributed Systems Security (NDSS) Symposium 2021.
[4] Chi, H., Zeng, Q., Du, X., Yu, J., 2020. Cross-app interference threats in smart homes: Categorization, detection and handling, in: 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE. pp. 411–423.
[5] Choi, J., Jeoung, H., Kim, J., Ko, Y., Jung, W., Kim, H., Kim, J., 2018. Detecting and identifying faulty iot devices in smart home with context extraction, in: 2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE. pp. 610–621.
[6] Fernandes, E., Jung, J., Prakash, A., 2016. Security analysis of emerging smart home applications, in: 2016 IEEE symposium on security and privacy (SP), IEEE. pp. 636–654.
[7] Fu, C., Zeng, Q., Chi, H., Du, X., Valluru, S.L., 2022. Iot phantom-delay attacks: Demystifying and exploiting iot timeout behaviors, in: 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), IEEE. pp. 428–440.
[8] Fu, C., Zeng, Q., Du, X., 2021. Hawatcher: Semantics-aware anomaly detection for appified smart homes, in: 30th {USENIX} Security Symposium ({USENIX} Security 21).
[9] Guan, Z., Lu, X., Wang, N., Wu, J., Du, X., Guizani, M., 2020. Towards secure and efficient energy trading in iiot-enabled energy internet: A blockchain approach. Future Generation Computer Systems 110, 686–695.
[10] Hnat, T.W., Srinivasan, V., Lu, J., Sookoor, T.I., Dawson, R., Stankovic, J., Whitehouse, K., 2011. The hitchhiker's guide to successful residential sensing deployments, in: Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems, pp. 232–245.
[11] Jakkula, V., Cook, D.J., 2008. Anomaly detection using temporal data mining in a smart home environment. Methods of information in medicine 47, 70–75.
[12] Sikder, A.K., Babun, L., Aksu, H., Uluagac, A.S., 2019. Aegis: a context-aware security framework for smart home systems, in: Proceedings of the 35th Annual Computer Security Applications Conference, pp. 28–41.
[13] SmartThings, 2017. its-too-cold.groovy. URL: https://github.com/SmartThingsCommunity/SmartThingsPublic/blob/master/smartapps/smartthings/its-too-cold.src/its-too-cold.groovy.
[14] SmartThings, 2021. Smartthings: One simple home system. a world of possibilities. URL: https://www.smartthings.com/.
[15] Statista, 2017. Smart home device household penetration in the united states in 2019 and 2021. URL: https://www.statista.com/statistics/1247351/smart-home-device-us-household-penetration/.
[16] Yamauchi, M., Ohsita, Y., Murata, M., Ueda, K., Kato, Y., 2020. Anomaly detection in smart home operation from user behaviors and home conditions. IEEE Transactions on Consumer Electronics 66, 183–192.
[17] Zhao, Y., Yu, Y., Li, Y., Han, G., Du, X., 2019. Machine learning based privacy-preserving fair data trading in big data market. Information Sciences 478, 449–460.