CP-IoT: A Cross-Platform Monitoring System for Smart Home

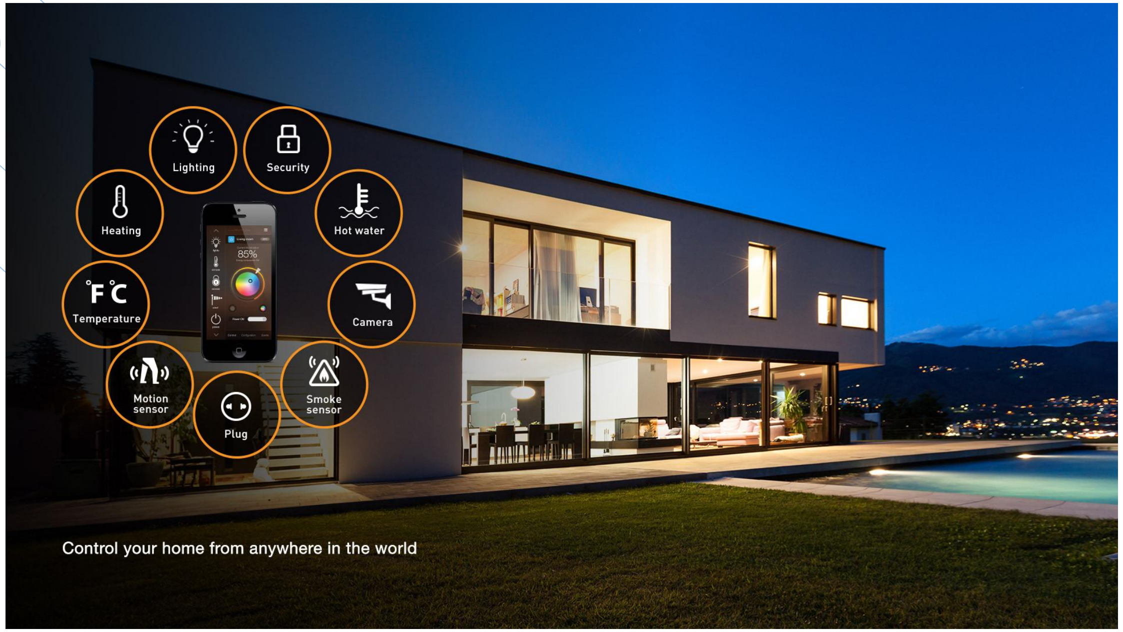
Hai Lin(Tsinghua University), Chenglong Li(Tsinghua University), Jiahai Yang(Tsinghua University), Zhiliang Wang(Tsinghua University), Linna Fan(Tsinghua University), Chenxin Duan(Tsinghua University, Alibaba Group)

NDSS SYMPOSIUM/2024

Presented by Internet Society

#NDSSSymposium2024

Internet of Things(IoT) is all around



Smart Homes

Smart Farms

Smart Healthcare

Smart Home facilitates our life

Smart Home Platforms

amazon alexa

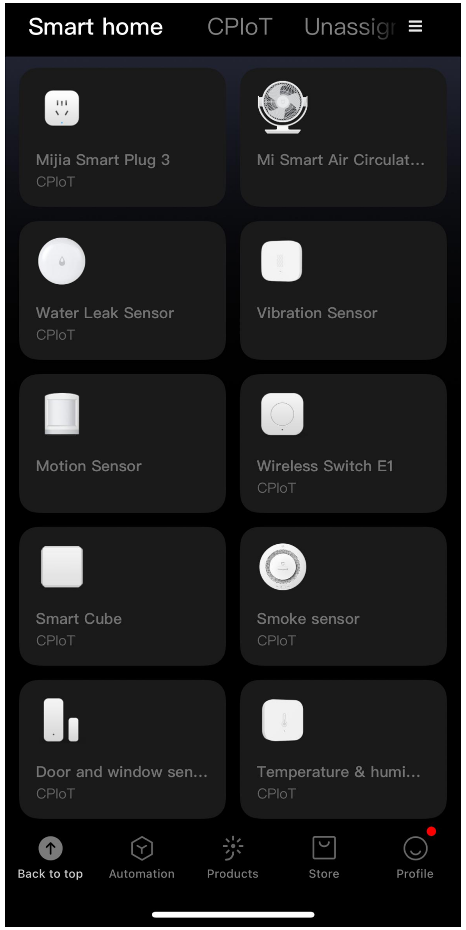
Nest

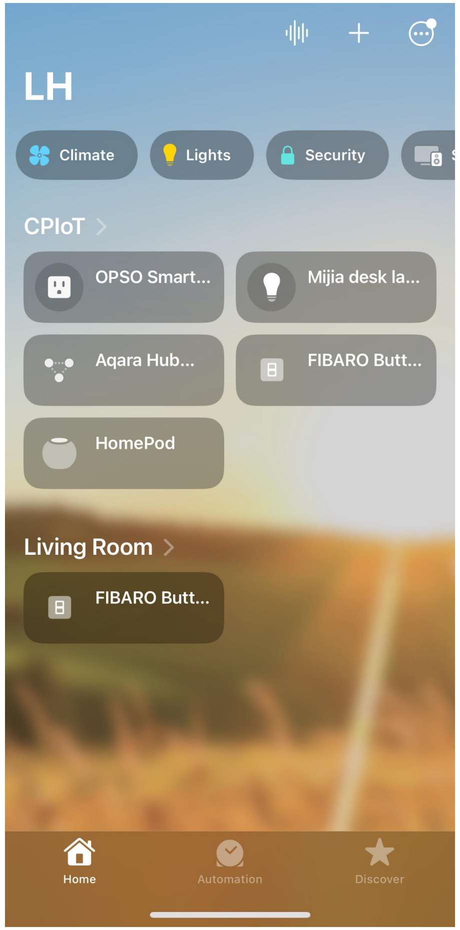
MiHome

Works with Apple HomeKit

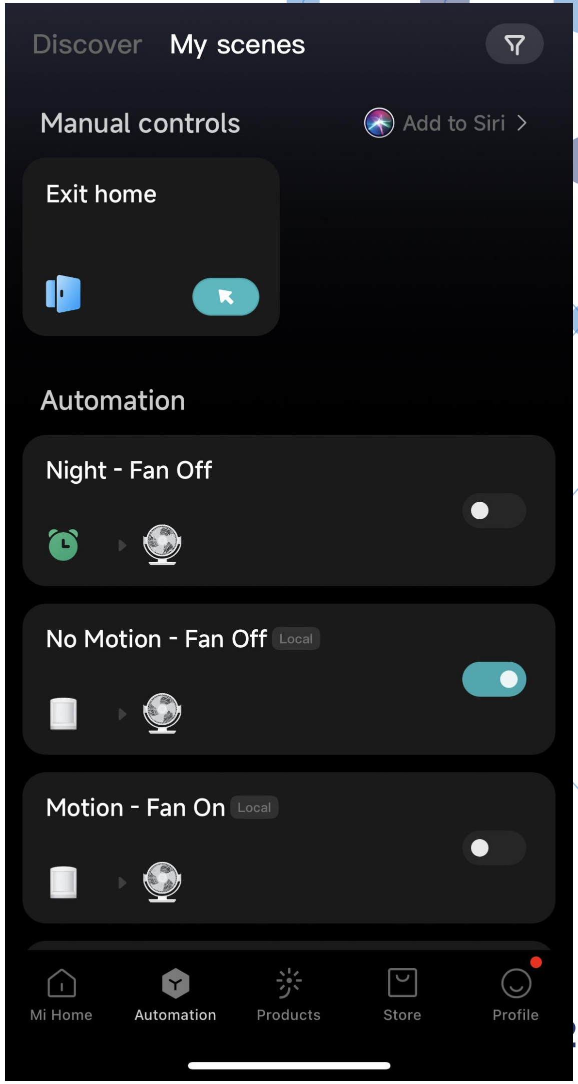
Works with SmartThings

Smart Applications

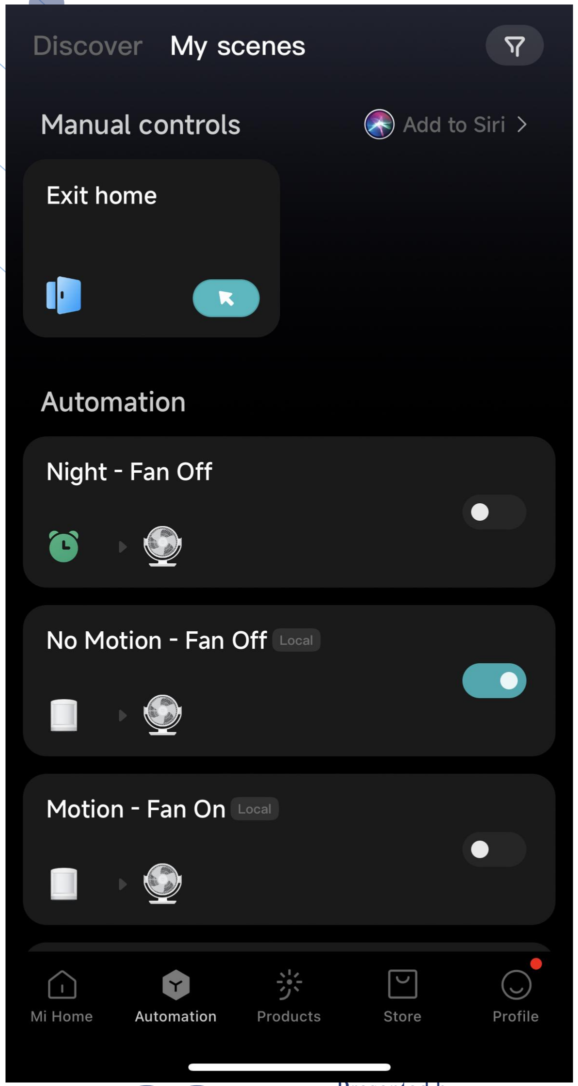




Automations



Home Automation Rule



ifthisthenthat

IF Event occurs THEN send Command
this that

Automation Rule: Night- Fan Off
E: Time at night(22 pm) C: Turn off the fan

$$E^{Time}_{time.night} \rightarrow C^{Fan}_{switch.off}$$

Automation Rule: No Motion- Fan Off
E: No motion detected C: Turn off the fan

$$E^{Sensor}_{motion.active} \rightarrow C^{Fan}_{switch.off}$$

Home automation faces various security problems

Automation is not trigging - Will run manually

Configuration automation

glen4cindy 2d

I've got a sensor for my garage door that reports open/closed conditions.

I've tried to set up an automation that will send a message if the door has been left open.

I'm pretty sure there must be something going on with how I have the automation configured because if I run it manually it works and sends the alert. If I just let things ride the automation never fires off even if I open the door and leave it open beyond the 10 minute threshold.

All of these problems caused by **attacks, device malfunctions, or faulty applications**

Sometimes lights turn off randomly

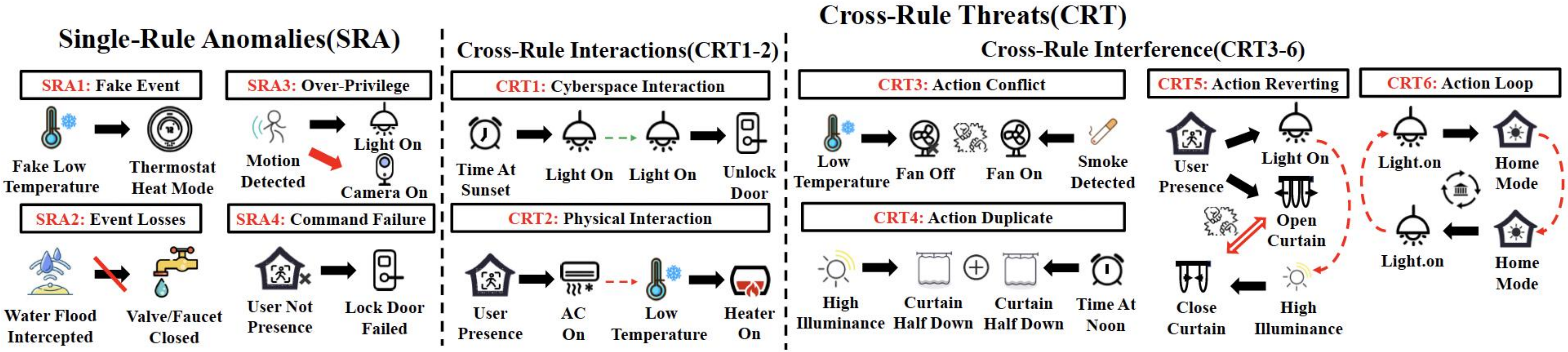
SmartApps & Automations

guessrow Jim Aug 2020

Hello all,

I have a couple GE smart switches that seem to turn off all by themselves at random times. All switches are 1-5 years old and I haven't mad any new routines lately, either. I do, however, have quite a few devices to control the switches (Action Tiles, Google Home, phones, tablets, etc.) Where do I start troubleshooting?

Various known threats types in home automation



Single-Rule Anomalies(SRA): Anomalies in automation rule execution

Cross-Rule Threats(CRT): Dangerous interactions and interferences across multiple rules. CRT also occur across multipe platforms.

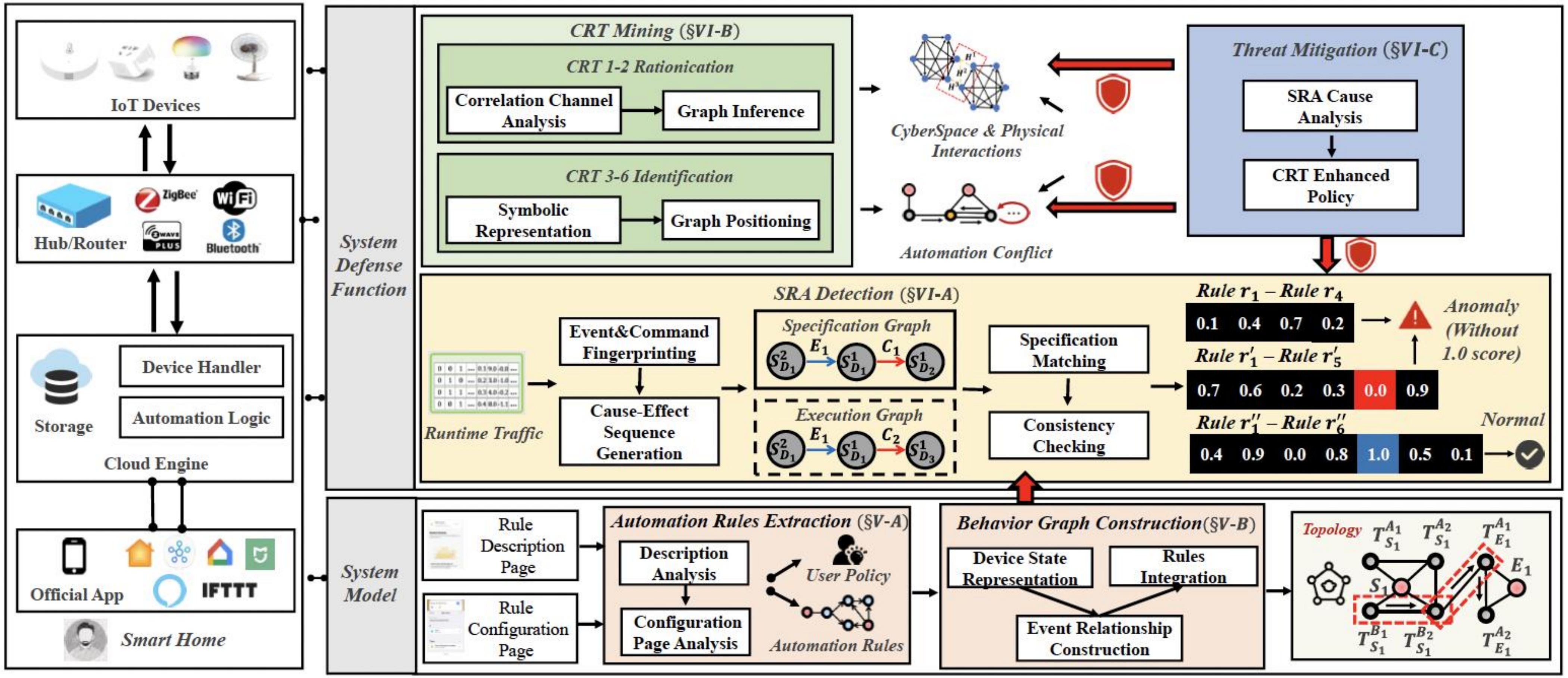
How can we detect all these threats
and support multiple smart home platforms?

Solution

We need a monitoring system for smart home to ...

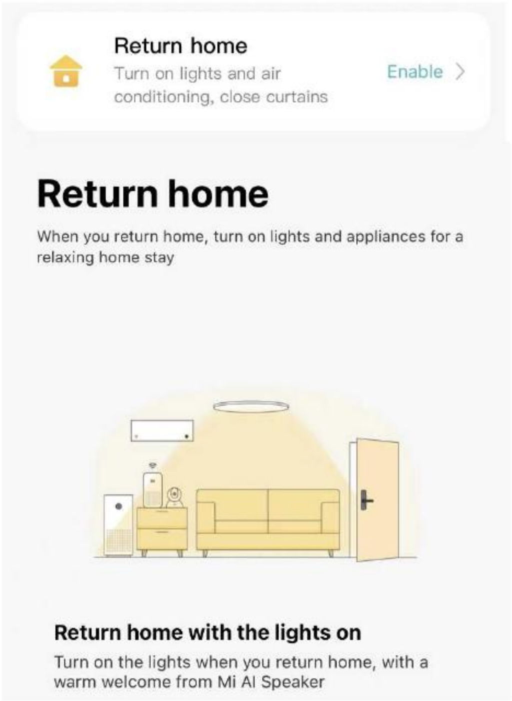
- **Extracting semantics** of automation rules from different platform apps
- **Modeling** devices and rules deployed on each platform
- **Identifying** the behavior of rule execution from the runtime environment and **detecting** anomalies
- **Mining** potential threats among various rules and **proposing** security policies to mitigate them

CP-IoT

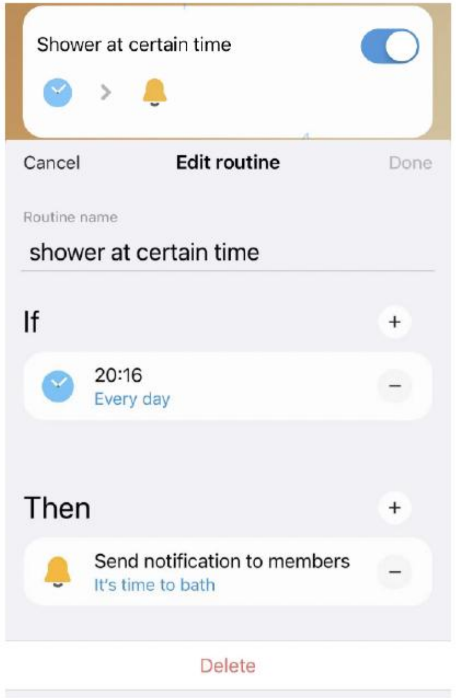


CP-IoT is a monitoring system for automation rule and device behaviors.
CP-IoT supports multiple smart home platforms.

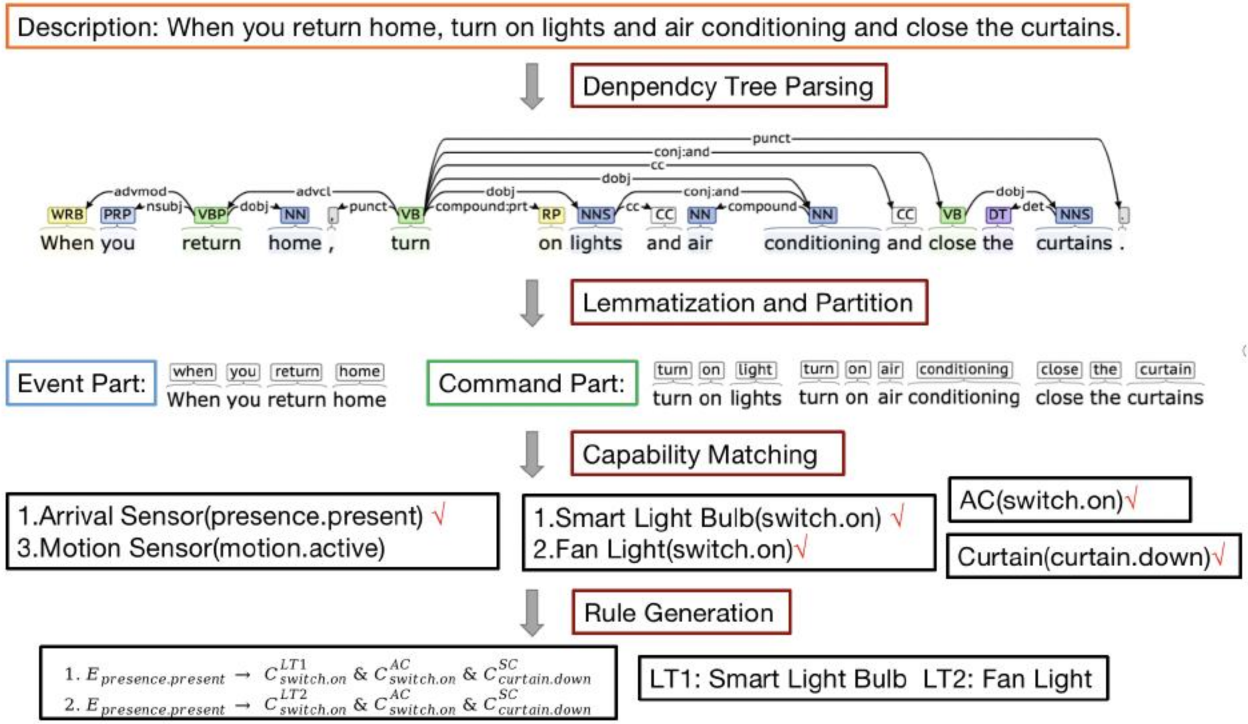
Automation rules extraction



Rule description page



Rule configuration page



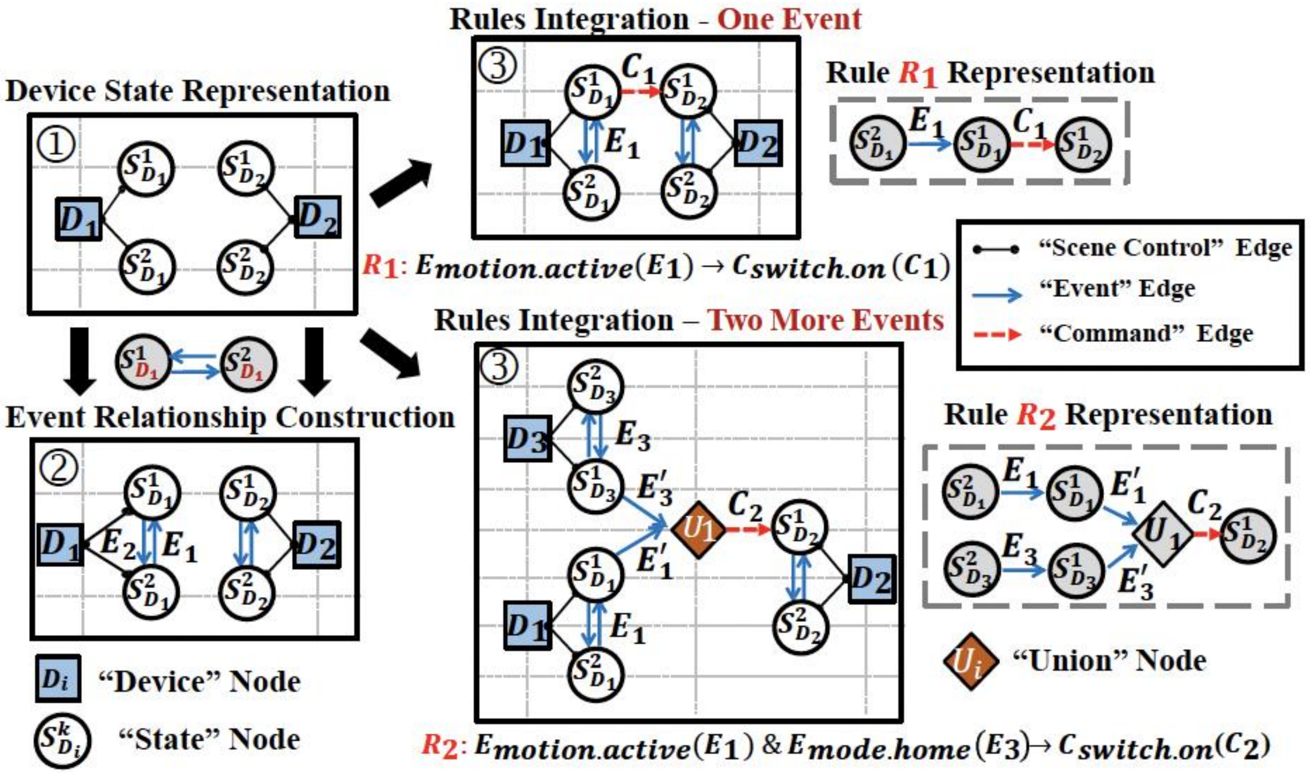
NLP Analysis

- **Two-stage analysis method on app pages**

➢ **First Stage:** Apply NLP analysis to collect semantics in rule description sentences.

➢ **Second Stage:** Extract user-defined parameters from the rule configuration page and supplement first-stage rule semantics.

Behavior graph construction



- **Building a centralized graph containing information from multiple platforms**

➢ Model device information and device state as **nodes**
➢ Model the **event** and **command** parts of a rule as **edges** between state nodes
➢ A state transfer chain represents the execution specification of a rule

Runtime Behavior Identification - Event&Command Fingerprinting

$$P = \{p_1, p_2, \dots, p_N\}$$

The traffic generated by a rule execution(N packets)

$$P = \{P_1, P_2, \dots, P_Q\} \quad s.t. \sum_{i=1}^{Q} |P_i| = N$$

Split the traffic into flows of events/commands

Constructing fingerprint for each event/command P_i (has s packets)

$$m_{P_i} = \begin{Bmatrix} p_{1,s_1} & p_{1,s_2} & p_{1,s_3} \\ p_{2,s_1} & p_{2,s_2} & p_{2,s_3} \\ \vdots & \vdots & \vdots \\ p_{s,s_1} & p_{s,s_2} & p_{s,s_3} \end{Bmatrix}$$

Packet-Level Fingerprint

$$f_{P_i} : (f_{i,1}, f_{i,2})$$

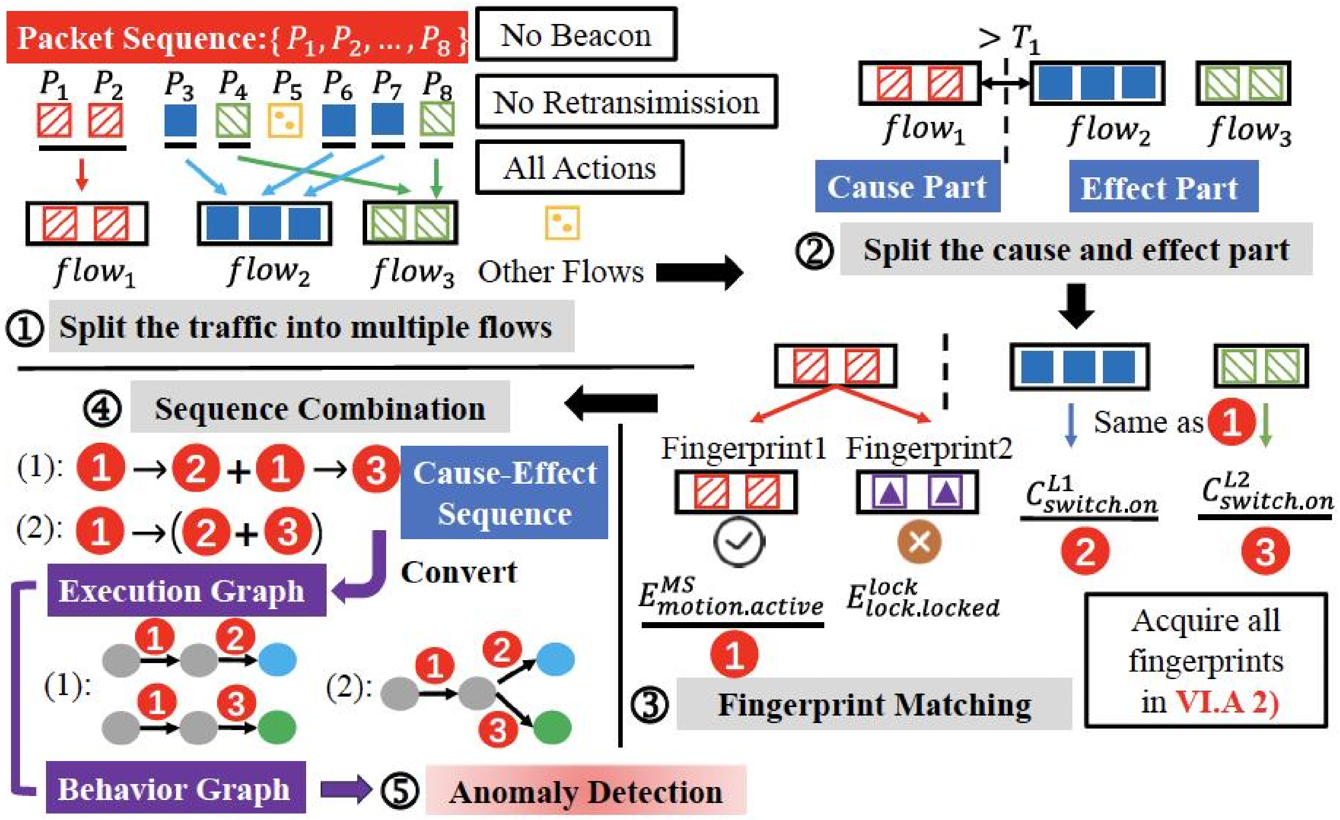
Flow-Level Fingerprint

Type	Statistic	Notation	Description
Packet	Size	s_1	Packet size will vary from event to event
	Protocol	s_2	WiFi(0), Z-Wave(1) Zigbee(2), Bluetooth(3)
	Direction	s_3	0: $device \rightarrow router$ 1: $router \rightarrow device$
Flow	Interval	f_1	Average packet interval
	Length	f_2	The length of packet sequence

Execute the rule T times and perform KMeans clustering on the resulting T event fingerprints

Eliminate the effects of **network latency** and **signal interference**

Runtime Behavior Identification - Cause-Effect Sequence Generation

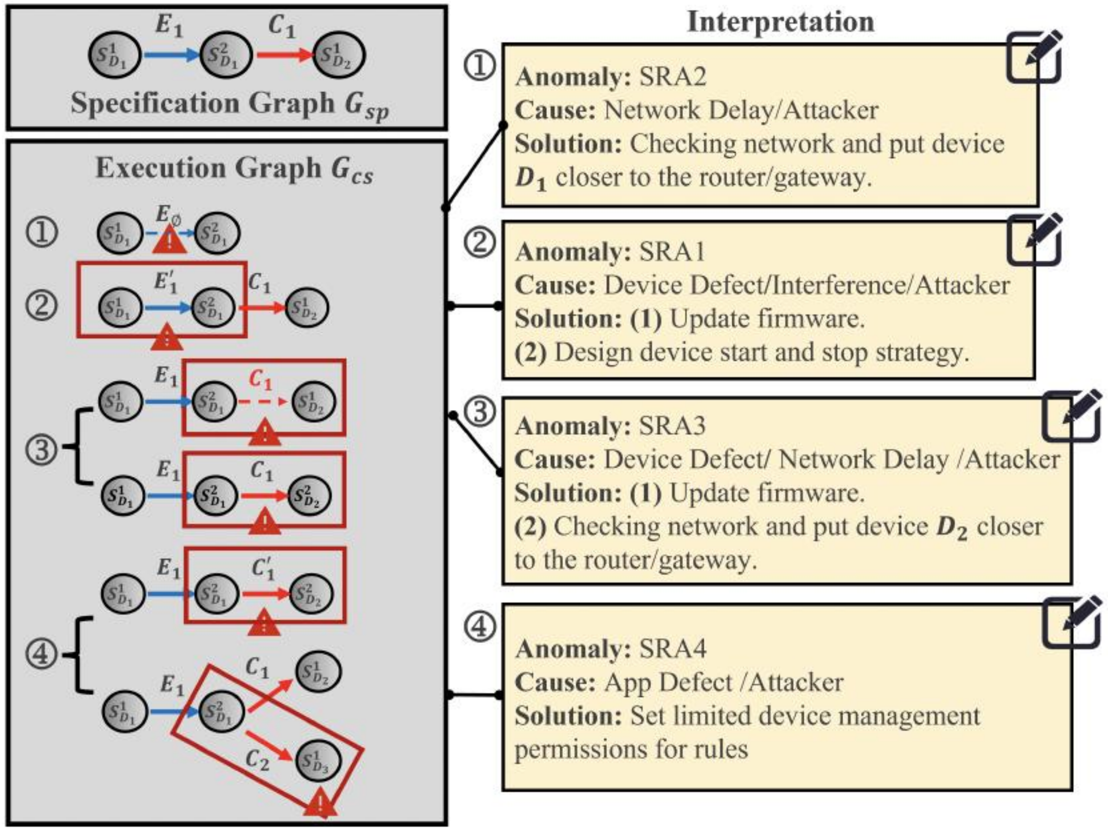


$$\underset{E_i/C_i}{argmin}\left(\underbrace{\lambda \cdot D\left(F_i,\ f_j\right)}_{Flow\ d_f}+\underbrace{\delta \cdot D\left(M_i, m_j\right)}_{Packets\ d_p}\right) \quad s.t.\ d_f+d_p \leq d$$

Calculate the **Manhattan Distance** and select the event/command that has the minimum weighted distance

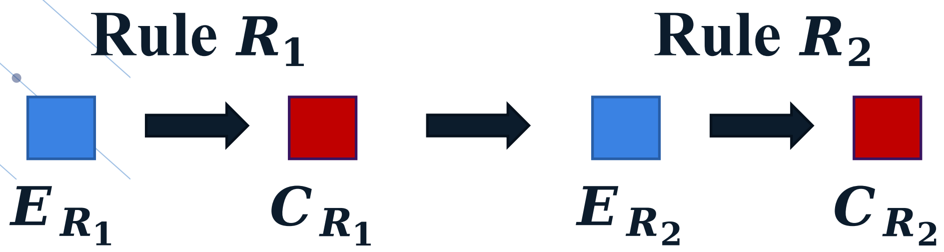
- Match runtime flow features($\boldsymbol{f_j}$, $\boldsymbol{m_j}$)with all fingerprints($\boldsymbol{F_i}$, $\boldsymbol{M_i}$) to identify which event occurs
- Combine multiple events and commands into cause-effect sequence based on dependencies
- Convert cause-effect sequence to the rule execution graph

Single-Rule Anomalies(SRA) detection



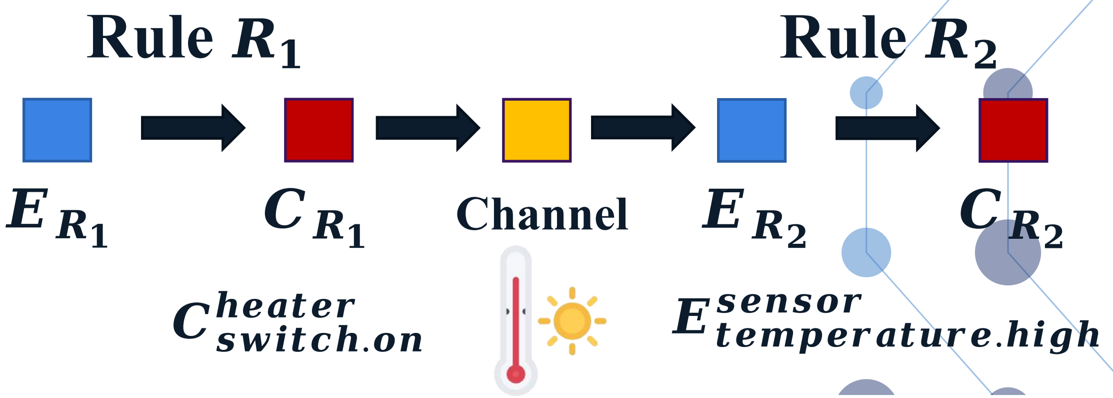
- **Specification Matching**
  - Locate the most similar part in the centralized graph based on all the events and commands contained in the execution graph.
- **Consistency Checking**
  - Calculate the similarity between two graphs based on node attributes, edge attributes and topology
  - Any inconsistency that occurs is an anomaly

Cross-Rule Interactions Discovery



- **Cyberspace Interactions**

  - The result of the execution of Rule $\boldsymbol{R_1}$ can directly trigger Rule $\boldsymbol{R_2}$
  - Constraints: $\boldsymbol{C_{R_1} \supseteq E_{R_2}}$



- **Physical Interactions**

  - The result of the execution of Rule $\boldsymbol{R_1}$ change the physical environment and indirectly trigger Rule $\boldsymbol{R_2}$
  - **Physical Correlation Analysis:** Applying the BERT model to calculate the **correlation scores** between the **command actions** of each rule and the 11 **physical channels**.
  - Constraints: $\boldsymbol{Corr(C_{R_1}) \supseteq E_{R_2}}$

NDSS SYMPOSIUM/2024

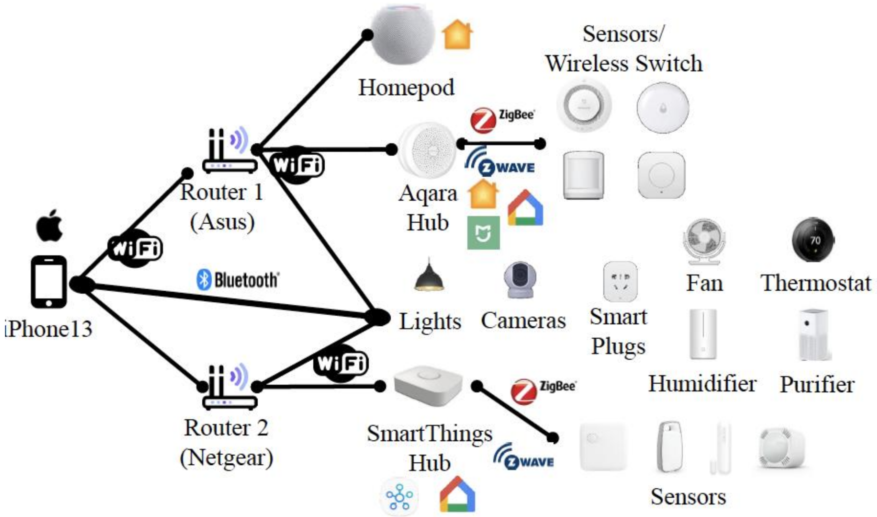
Presented by Internet Society

Cross-Rule Interferences Identification

Type	Representation
Action Conflict CRT3	$\exists q_1 \in C_{r1}, q_2 \in C_{r2}$ $S_1^s = q_1.suc, S_2^s = q_2.suc$ $s.t.\ \ S_1^s.cp = S_2^s.cp, S_1^s.val \neq S_2^s.val$
Action Duplicate CRT4	$\exists q_1 \in C_{r1}, q_2 \in C_{r2}$ $S_1^s = q_1.suc, S_2^s = q_2.suc$ $s.t.\ \ S_1^s.cp = S_2^s.cp, S_1^s.val = S_2^s.val$
Action Reverting CRT5	$\exists q_1 \in C_{r1}, q_n \in C_{rn}$ $S_1^s = q_1.suc, S_n^s = q_n.suc$ $s.t.\ \ S_1^s.cp = S_2^s.cp, S_1^s.val \neq S_n^s.val,$ $\forall_{i=1}^{n-1}(r_i, r_{i+1}) \in S_{cyb}/S_{phy}$
Action Loop CRT6	$s.t.\ \ E_{r1} \subseteq C_{rn}$ $\forall_{i=1}^{n-1}(r_i, r_{i+1}) \in S_{cyb}/S_{phy}$

- **Symbolic Representation**
  - Representation of various interference types as constraints on graphs based on their semantics.
- **Graph Positioning**
  - For action conflict and action duplicate, find **two rules** on the graph that satisfy the constraints.
  - For action reverting and action loop, find **two more rules** on the graph that satisfy the constraints.

Testbed



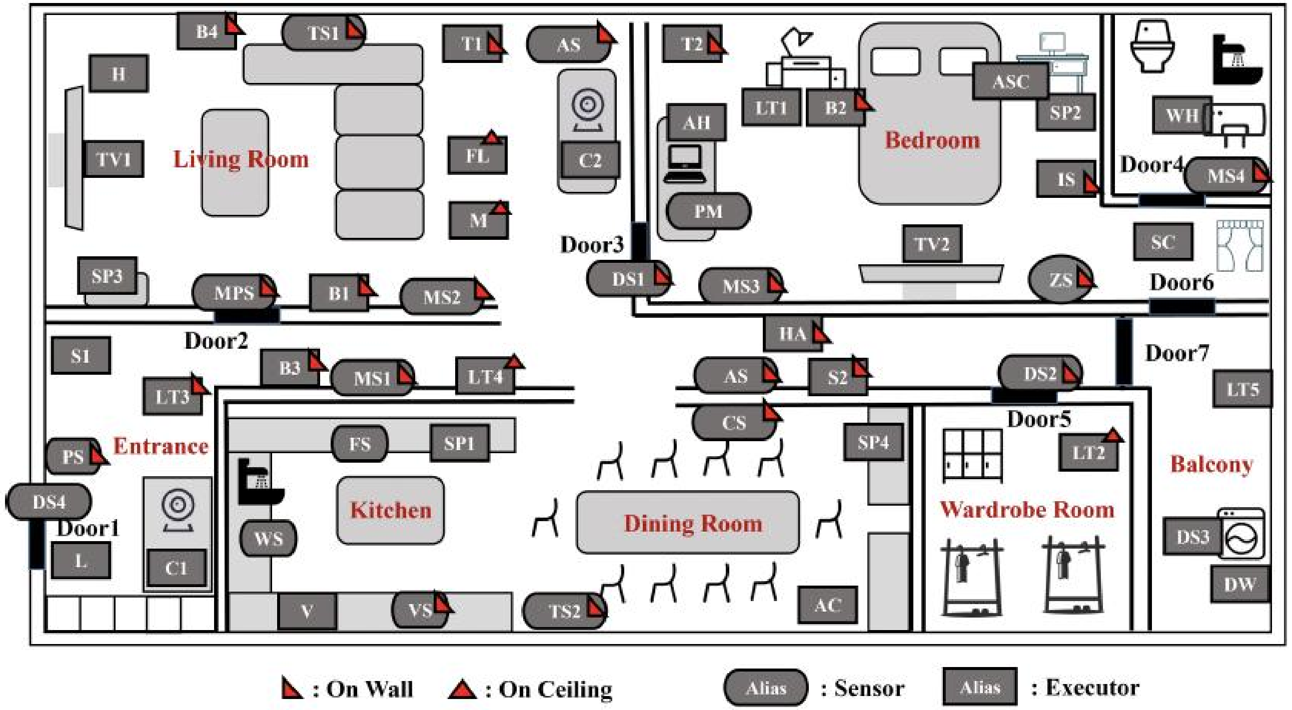
- **Real Testbed**

➢ 32 IoT devices and 4 platforms: Homekit, SmartThings, Google Home, Xiaomi Home

➢ Automation rules: SmartThings(105), Homekit(128), Google Home(160), Xiaomi Home(192)

➢ Anomalies： Each rule injects four anomaly types

Testbed



- **Simulation Testbed**

➢ **54 IoT devices**

➢ **2491 automation rules**: Crawl 10796 applets from the IFTTT website and 82 SmartApp from the SmartThings Public GitHub Repository, and filter 2491 rules associated with these devices.

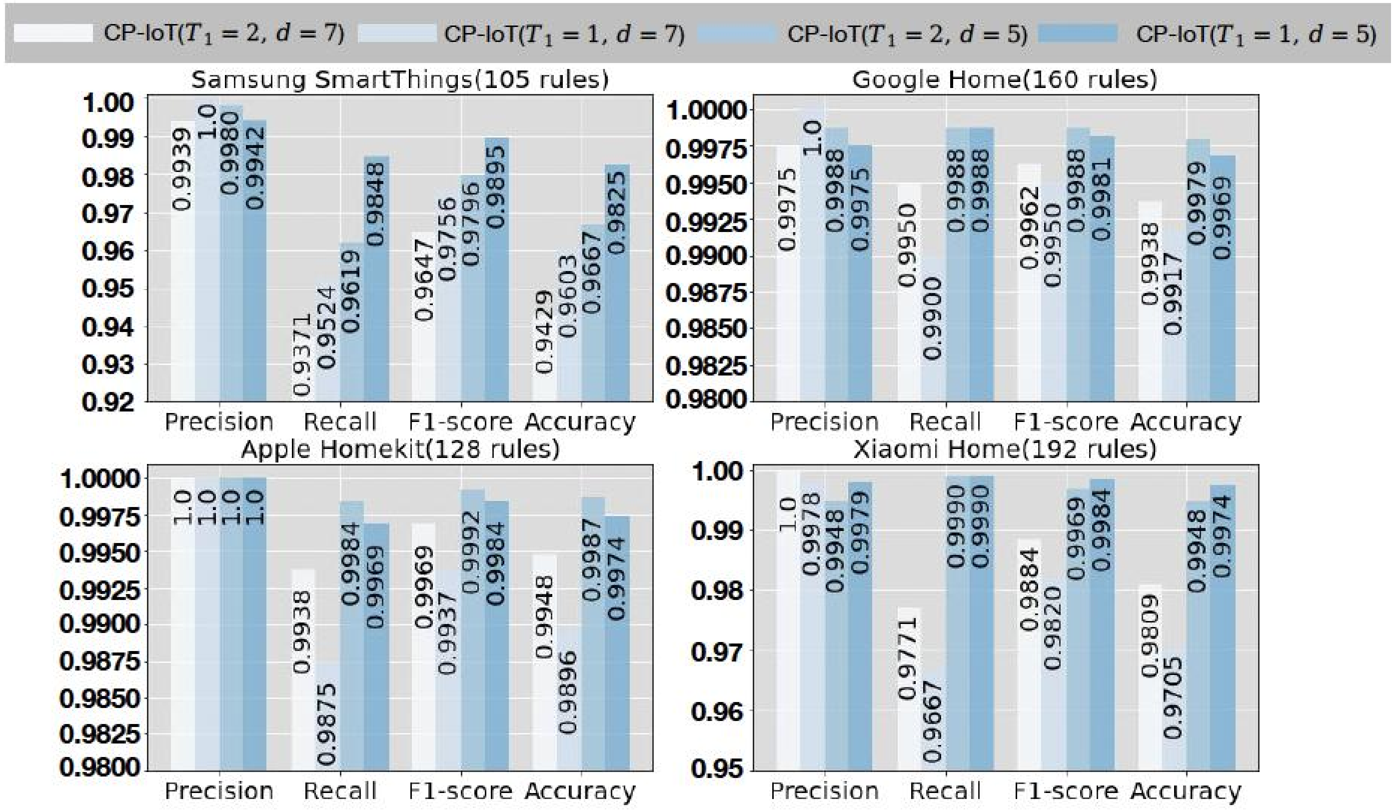
Rule Extraction Accuracy

Platform	Word2Vec [47]		BERT [38]	
	DAnalysis	+CAnalysis	DAnalysis	+CAnalysis
SmartThings(105)	89.52%	92.38%	91.43%	97.14%
Apple Homekit(128)	89.06%	96.09%	92.97%	99.22%
Google Home(160)	83.13%	95.63%	91.25%	98.13%
Xiaomi Home(192)	85.94%	91.15%	88.02%	98.96%

DAnalysis: Description Page Analysis
CAnalysis: Configuration Page Analysis

- Using **Word2vec** or **BERT** for device capability matching and **BERT outperform Word2vec.**
- BERT average accuracy higher than **98.9%**
- A percentage of the description statements are ambiguous, and the accuracy of rule semantic extraction can be improved by configuration analysis

Anomaly Detection Performance



$\boldsymbol{T_1}, \boldsymbol{d}$: Predefined parameters
$\boldsymbol{T_1}$: Time interval of dependencies
$\boldsymbol{d}$: Allowable error range for fingerprint matching

- **Average precision:** higher than 99.0%
- **Average recall of CP-IoT with the best configuration:** higher than 98.0%.

- **False Negative Causes:** (1) Small packet deviation (2) Fail to get the log information

(3) SmartThings have high response latency and may be caused by proxy servers.

Cross-Rule Threats Discovery Performance

No.	Rule1	Rule2	Type	Testbed	Risk	No.	Rule1	Rule2	Type	Testbed	Risk
1	$E^{sensor}_{presence.present} \rightarrow C^{fan}_{switch.on}$	$E^{sensor}_{motion.active} \rightarrow C^{light}_{switch.on}$	P	R	Low	9	$E^{sensor}_{motion.active} \rightarrow C^{light}_{switch.on}$	$E^{sensor}_{illuminance.high} \rightarrow C^{curtain}_{windowShade.close}$	P	S	Low
2	$E^{sensor}_{vibration.active} \rightarrow C^{humidifier}_{switch.on}$	$E^{sensor}_{water.wet} \rightarrow C^{light}_{color.blue}$	P	R	Low	10	$E^{sensor}_{CO.detected} \rightarrow C^{siren}_{alram.siren}$	$E^{sensor}_{sound.high} \rightarrow C^{TV}_{volume.down}$	P	S	Low
3	$E^{sensor}_{motion.active} \rightarrow C^{humidifier}_{switch.on}$	$E^{sensor}_{humidity.high} \rightarrow C^{fan}_{switch.on}$	P	R	Low	11	$E^{sensor}_{dustLevel.high} \rightarrow C^{robot}_{switch.on}$	$E^{sensor}_{motion.active} \rightarrow C^{dishwasher}_{switch.on}$	P	S	High
4	$E^{sensor}_{smoke.detected} \rightarrow C^{fan}_{switch.on}$	$E^{sensor}_{temperature.low} \rightarrow C^{thermostat}_{mode.heat}$	P	R	High	12	$E^{sensor}_{humidity.low} \rightarrow C^{humidifer}_{switch.on}$	$E^{powerMeter}_{energy.high} \rightarrow C^{camera}_{switch.off}$	P	S	High
5	$E^{button}_{button.pressed} \rightarrow C^{thermostat}_{mode.heat}$	$E^{sensor}_{temperature.high} \rightarrow C^{fan}_{switch.on}$	P	R	Low	13	$E^{sensor}_{contact.closed} \rightarrow C^{TV}_{switch.off}$	$E^{sensor}_{sound.low} \rightarrow C^{camera}_{switch.on}$	P	S	High
6	$E^{sensor}_{temperature.low} \rightarrow C^{fan}_{switch.off}$	$E^{sensor}_{motion.inactive} \rightarrow C^{camera}_{switch.on}$	P	R	High	14	$E^{sensor}_{dustLevel.high} \rightarrow C^{robot}_{switch.on}$	$E^{sensor}_{presence.present} \rightarrow C^{lock}_{lock.unlocked}$	P	S	High
7	$E^{sensor}_{contact.open} \rightarrow C^{light}_{switch.on}$	$E^{light}_{switch.on} \rightarrow C^{light}_{color.blue}$	C	R	Low	15	$E^{sensor}_{illuminance.low} \rightarrow C^{light}_{switch.on}$	$E^{light}_{switch.on} \rightarrow C^{curtain}_{windowShade.open}$	C	S	Low
8	$E^{Time}_{time.night} \rightarrow C^{Mode}_{mode.night}$	$E^{Mode}_{mode.night} \rightarrow C^{camera}_{switch.on}$	C	R	High	16	$E^{sensor}_{motion.active} \rightarrow C^{Mode}_{mode.home}$	$E^{Mode}_{mode.home} \rightarrow C^{window}_{window.open}$	C	S	High

Found some typical cross-rule interactions

Number / Kind \ Method	IoTGaze	iRuler	CP-IoT
Physical Interactions(CRT1)	827	N/A	1461
Cyberspace Interactions(CRT2)	344	N/A	1072
Action Conflict(CRT3)	N/A	4619	4723
Action Duplicate(CRT4)	N/A	6025	6108
Action Reverting(CRT5)	N/A	2704	2855
Action Loop(CRT6)	N/A	1856	2039

- **Find more cross-rule interactions:** CP-IoT considers more channels and rules combination
- **Slightly more cross-rule interferences**: both CP-IoT and iRuler perform a complete searching of the rule combination space, but CP-IoT considers more feasible rule chains triggered by multiple physical interactions.

Thanks for listening!

colinLH

linhai17181@gmail.com

linhaiwebsite.cn