EB(B) EBB

e2s

5.4 Correlation Refining

(),where

The accepted hypothetical correlations should not be used

b', resp.) are values of the attribute resp.)

directly for two reasons. First, conditions of smart apps may

after numeric-to-binary conversion; symmetrically, we gen-

be overlooked if they remain unchanged during training.

erate another eight hypothetical correlations with the events

For instance, assume there is a smart app that, upon the

of B as anteriors.

front door opening, turns on the porch light after sunset.

Moreover, we propose to combine semantics from smart

If the residents always come back home after sunset, the

apps with semantics from the adjacency table. The intuition

inaccurate correlation (Econtact

switch(PorchLight)

Eon

could

behind the combination is that when an action command in a

be accepted by hypothesis testing and cause false alarms of

smart app is executed, it usually imposes certain changes on

"porch light not turned on" when the residents return before

one or more attributes. Given an e2e correlation containing

sunset. Second, when apps change, accepted hypothetical

a condition extracted from a smart app, we create a virtual

correlations may become outdated and contradict with the

device, which reports an event when both the trigger event

e2e correlations newly derived from apps. This can also cause

arises and the condition is true. For instance, a virtual mo-

false alarms, as confirmed by our experiments (Section 6.5).

tion sensor is created according to the conditional trigger

-Motion(M) \Spresence(PS) which becomes active only when

We thus propose to refine mined correlations using e2e

correlations extracted from smart apps, and launch the re-

Factive arises and PS is present. Next, the virtual device

fining process whenever smart app changes or there are

is used, just like the corresponding real device, to generate

hypothetical correlations accepted by hypothesis testing.

hypothetical correlations according to the adjacency table.

We first define the cover relation between two correlations:

Our current prototype only considers devices installed

an e2e correlation extracted from a

in the same room for generating hypothetical correlations.

smart app covers a correlation that

EDID)

While this can be relaxed by considering any two devices

passes hypothesis testing if they meet two conditions: 1)

in the home, our current implementation makes a trade-off

they have the same posterior event (i.e., EB(B) and

between the comprehensiveness of hypothetical correlations

Ed(A) (logically) implies EY(C)

If

and the meaningfulness of the mined correlations.

Cs covers Ch, the latter is removed. In the example men-

5.3.3 Hypothesis Testing

tioned above, a smart app derived e2e correlation (Econtact ^

glocation

witch(PorchLight)

It is worth emphasizing that hypothetical correlations are not

sunset

Eon

covers the mined correlation

necessarily true. That is why we need hypothesis testing, the

(Econtact

Eon

switch(PorchLight)

because they have the same

process of verifying hypothetical correlations using event

posterior event and (Econtact glocation Econtact Popen thus,

the

logs. Given a hypothetical correlation, we traverse event logs

latter correlation is removed.

to find all events that match its anterior, and take each of them

as a testing case. Then, we check whether the hypothetical

5.5 Anomaly Detection

correlation's posterior event or state is consistent with the

SmartThings does not provide access to its internal content,
physical ground truth as recorded in event logs. For example,
an event instance of FMotion active constitutes a testing case for
such as device states. To overcome the barrier, we design a
shadow execution engine, which subscribes to the events of
the hypothetical correlation ESwitch(Light)). This
the installed IoT devices. It keeps track of all devices' states
case is counted as a success if Eon switch(Light) occurs within a
and simulates a smart home's legitimate behaviors based on
short duration d after Motion active In our implementation, d =
obtained correlations.
60s, which is long enough to wait for the feedback event to
For each incoming event, the shadow execution engine
arrive but not too long as to accept an event not related to
performs the Contextual and Consequential checking succes-
Motion Factive Note the scheduling granularity of SmartThings is
sively. The contextual checking verifies whether the event
at per-minute level [1].
occurs in a valid context specified in e2s correlations. After
Checking these testing cases can be considered as a se-
that, the consequential checking searches for its consequen-
quence of independent Bernoulli trails. We use the one-tail
tial events as predicted by e2e correlations.
test [42] to evaluate each hypothetical correlation's correct-
Below, we use the same example correlation (between a
ness. For a given correlation, we set the alternative hypothe-

motion sensor and a light) as in Section 4.2. When an event

sis H as "the correlation succeeds with a probability higher

-Motion(A)

Eactive

is received, the shadow execution engine first con-

than P0". Correspondingly, the null hypothesis H is "the

ducts the contextual checking. It traverses all e2s correlations

correlation succeeds with a probability no higher than P0".

and locates those with the event Eactive ,Motion(A) at their anterior

We choose the 95% fiducial probability as in common prac-

places. Among the located e2s correlations, if any of them

tices [27], which means that the correlation can only be

have states in their posterior places that are inconsistent

accepted if the null hypothesis's p-value is smaller than 5%.