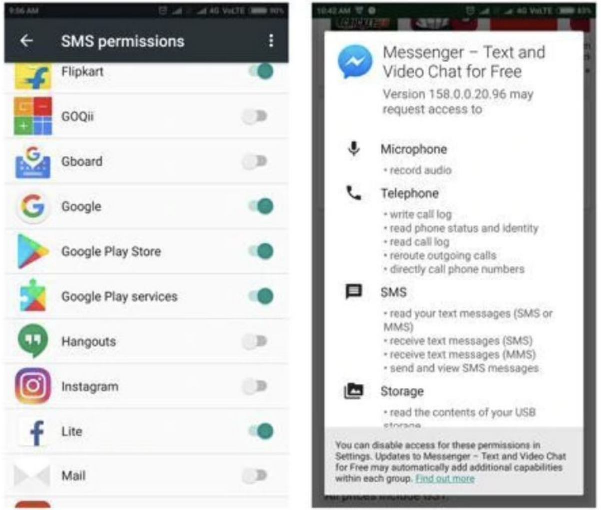
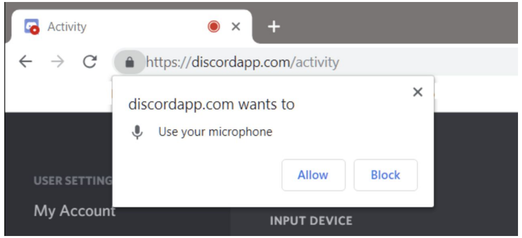
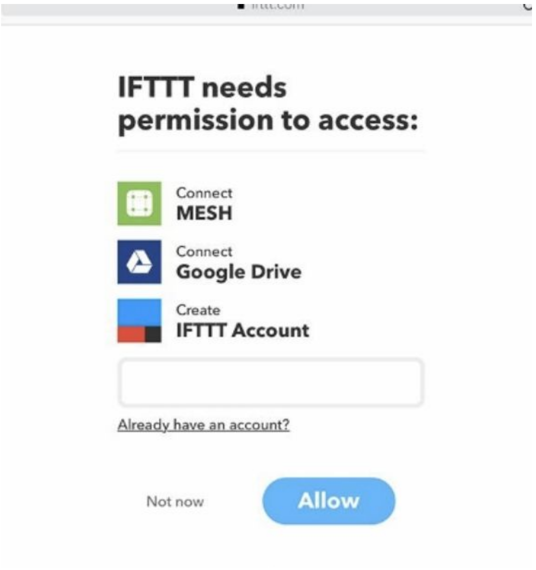
TKPERM: Cross-platform Permission Knowledge Transfer to Detect Overprivileged Third-party Applications

Faysal Hossain Shezan and **Kaiming Cheng** (University of Virginia);
Zhen Zhang and Yinzhi Cao (Johns Hopkins University);
Yuan Tian (University of Virginia)

Permission-based access control







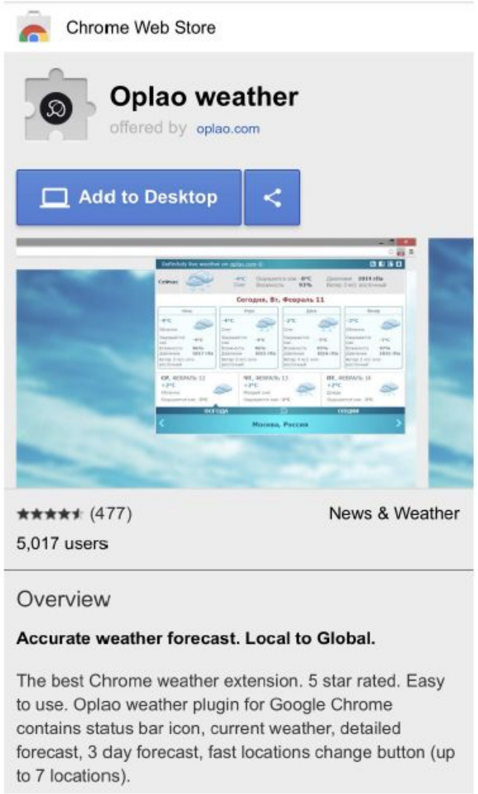
Android

Chrome

IFTTT

Case Study

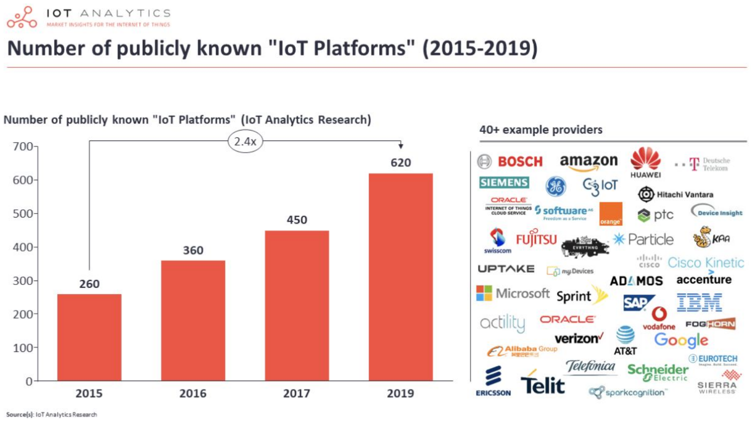
Bridging the gap between user’s expectation and app behavior



Challenge

Extensive data labeling and parameter tuning on new platforms

Source code is often unavailable

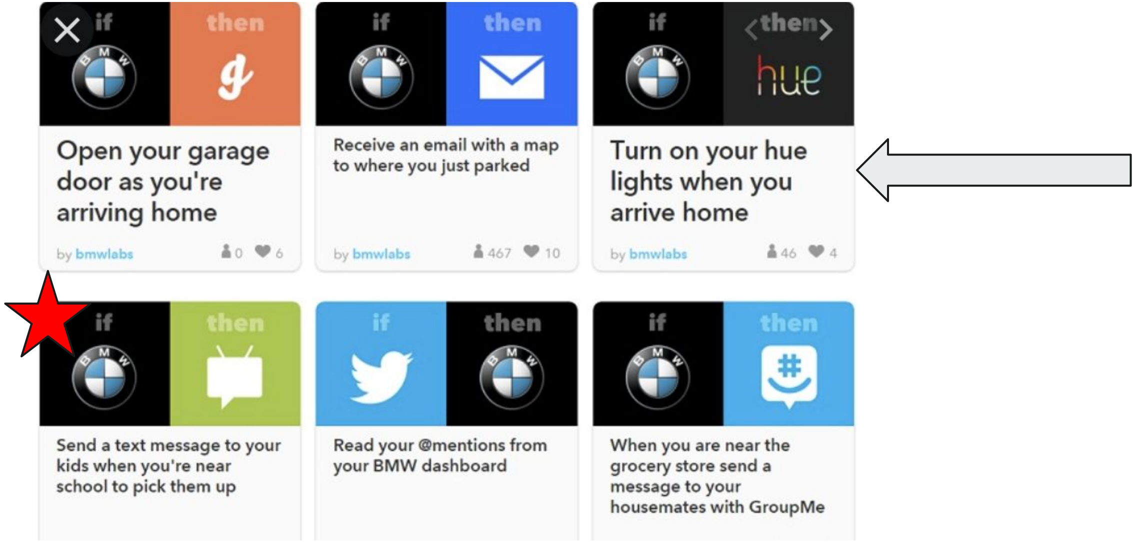


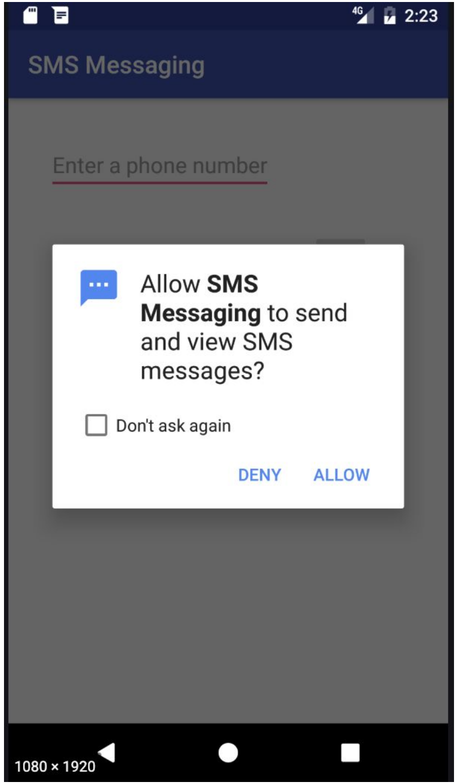
Reference:https://iot-analytics.com/iot-platform-companies-landscape-2020/
https://users.cs.northwestern.edu/~ychen/Papers/CCS14.pdf
https://www.usenix.org/conference/usenixsecurity13/technical-sessions/presentation/pandita

Key Insight

While these platforms are varied with different use cases, or have different sets of permissions, they are **all user-facing,** thus sharing certain aspects that are **transferable** across platforms.

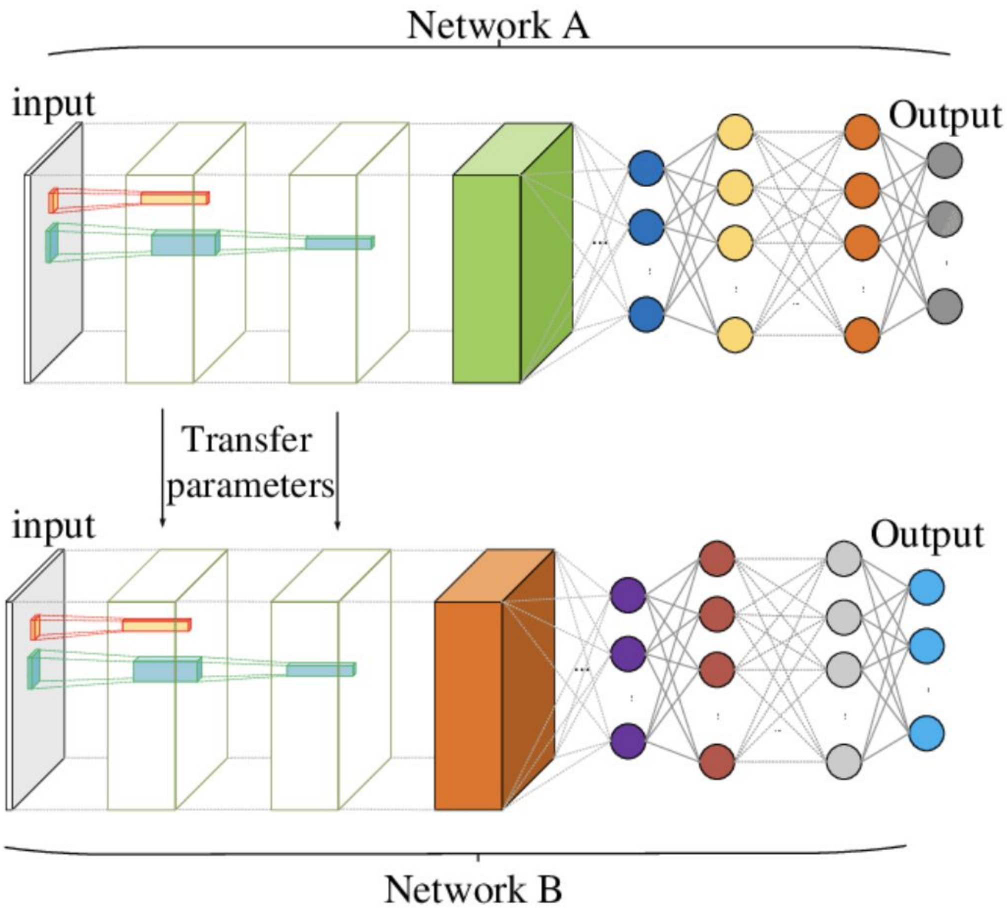
Example


if
then
Open your garage door as you're arriving home
by bmwlabs
0
6
if
then
Receive an email with a map to where you just parked
by bmwlabs
467
10
if
then
hue
Turn on your hue lights when you arrive home
by bmwlabs
46
4
if
then
Send a text message to your kids when you're near school to pick them up
if
then
Read your @mentions from your BMW dashboard
if
then
When you are near the grocery store send a message to your housemates with GroupMe

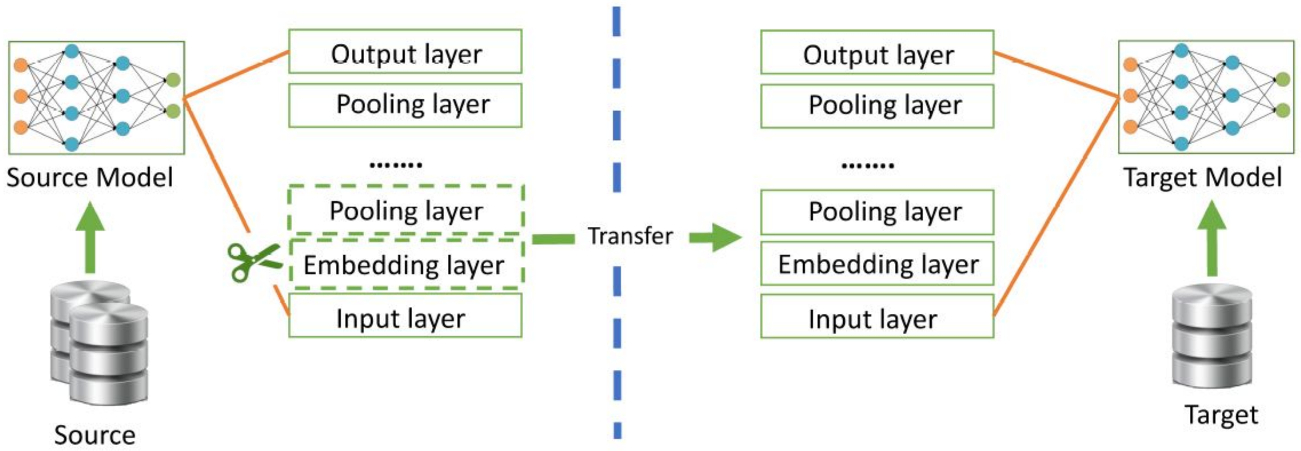

2:23
SMS Messaging
Enter a phone number
Allow **SMS Messaging** to send and view SMS messages?
Don't ask again
DENY
ALLOW
1080 × 1920

Background

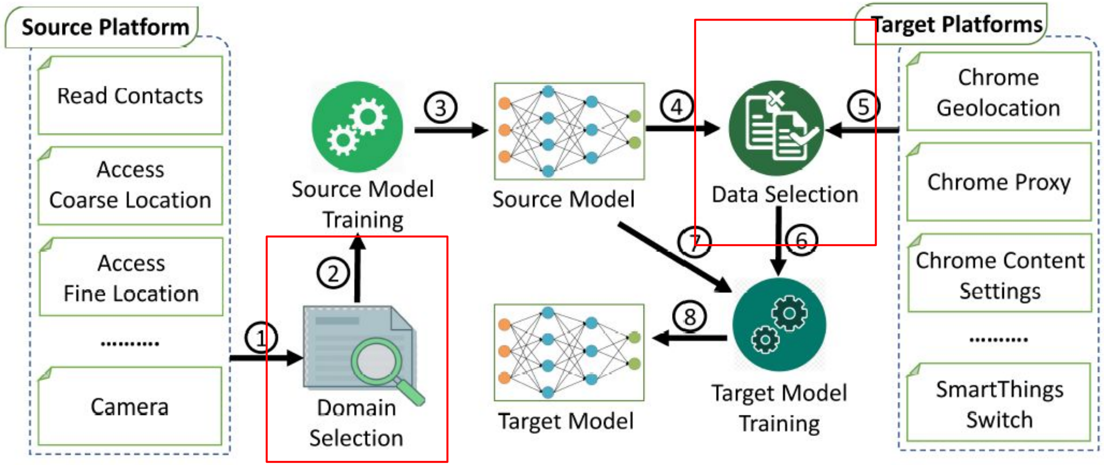
Transfer learning (TL) is a research problem in machine **learning** (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem



Solution - Transfer Learning



System Overview



Implementation - Dataset

Android：Adopted the crawled data, provided by the authors of Autocog

Chrome Extension: We build a **Chrome data crawler** to get all the application’s information.

IFTTT: We collected 259,523 IFTTT recipes in October 2017 using our crawler built with python and **beautiful soup**.

SmartThings: We collected 243 SmartThings applications in August 2019.

Dataset - cont'd

- What is our labeling process
- How to handle disagreement？（agreement rate as 97.89%）
- **Example：**
- ***"When you have a meeting, auto create a note at Evernote"***, which belongs to an IFTTT recipe requiring access to Google Calendar.

Dataset - cont’d

- What is our labeling process
- How to handle disagreement？（agreement rate as 97.89%）
- **Example：**
- ***“When you have a meeting, auto create a note at Evernote”***, which belongs to an IFTTT recipe requiring access to Google Calendar. Two annotators have disagreement because one thinks that this sentence has no relationship with Google Calendar, while the other thinks that a recipe can only know that you have a meeting based on an access to Google Calendar.

Implementation - Dataset

Plat.	Permission	#Sent.	#Pos. Sent.	#Doc.	#Pos. Doc.
Android	Fine Loc.	16,402	728 (4.44%)	635	635 (100%)
	Coarse Loc.	5,550	208 (3.75%)	193	193 (100%)
	Camera	498	166 (33.33%)	11	11 (100%)
	Read Cal.	802	401 (50.00%)	16	16 (100%)
	Read Con.	842	421 (50.00%)	17	17 (100%)
	Record Au.	366	183 (50.00%)	10	10 (100%)
	Wr. Settings	1,524	398 (26.12%)	31	31 (100%)
	Send SMS	8,398	407 (4.85%)	286	286 (100%)
	Write APN	1,811	92 (5.10%)	35	35 (100%)
IFTTT	Evernote	202	133 (65.84%)	145	85 (58.6%)
	BMW Lab	77	52 (67.53%)	65	43 (66.2%)
	Facebook	158	84 (53.16%)	115	45 (39.1%)
	G. Cal.	144	88 (61.11%)	102	73 (71.6%)
	G. Con.	85	50 (58.82%)	49	43 (87.8%)
Chrm.	Geoloc.	1,540	126 (8.18%)	138	67 (48.6%)
	Proxy	2,391	483 (20.20%)	123	98 (79.7%)
	C. Settings	774	92 (11.89%)	58	28 (48.3%)
Smart Things	Lock	34	10 (29.31%)	30	8 (26.67%)
	Motion	73	40 (54.79%)	60	35 (58.33%)
	Switch	185	118 (63.78%)	153	111(72.55%)

In total, we labeled **36,193** sentences from 1,234 Android applications, **666** sentences from 476 IFTTT recipes, **4,705** sentences from 319 Chrome extensions and **292** sentences from 243 SmartThings applications.

https://drive.google.com/open?id=1cEZ4MiolsbV4fXaDyJsUtHDGoPr8StjM”

Password: 6eZPq2h”.

Models & Hypermeter

The instance we used is called ‘p3.2xlarge’ with one NVIDIA Tesla V100 GPU, 16 Gibibyte GPU memory, 8 virtual central processing units (vCPUS) and 61 Gibibyte Main Memory. The operating system of this instance is the ‘Deep Learning Amazon Linux Version 23.0’.

Learning rate = 0.01
Batch size = 256
Number of Epoch = 20
Rank size = 20

Algorithm & Application

- Adopts **CBoW (Continuous Bag-of- Words)** encoder to translate each sentence into a vector
- TKPERM pre-processes all the sentences by following the standard NLP practice, such as removing Unicode character, punctuation, stop words, etc
- Choose **FCNN (Fully Connected Neural Network)** for building our model structure for source domain knowledge distilling (Compared with LSTM)

Challenge -- How to handle unique permission

- Given that we have 9 different source domain, brute-forcing will occur 2^9 possibilities.
- State-of-the art domain selection technique doesn’t output desired outcome. (H-Divergence)
- What is our solution and our takeaway from that？
- Discussion.

Challenge -- How to handle unique permission

Algorithm 1 Source Domain Selection using Greedy Selection Algorithm

Input: Source Domain Data List, $[\mathcal{D}_S]$; Target Domain Data, d_t

Output: Aggregated Source List, $[\mathcal{A}_S]$

procedure SELECTSOURCEDOMAINS
    $[\mathcal{A}_S] \leftarrow \emptyset$
    $P_{best} \leftarrow -\infty$
    $P_{current} \leftarrow$ initialize to $zero$
    $[\{D_S, d_{f1}\}] \leftarrow computeallds_{f1}([D_S], d_t)$
    while $size([\{D_S, d_{f1}\}]) > 0$ **do**
        $d_s \leftarrow highest_{f1}([\{D_S, d_{f1}\}])$
        remove d_s from $[\{D_S, d_{f1}\}]$
        add d_s to $[\mathcal{A}_S]$
        $P_{current} \leftarrow computeds_{f1}([\mathcal{A}_S], d_t)$
        if $P_{current} < P_{best}$ **then**
            remove d_s from $[\mathcal{A}_S]$
            break
        end if
        $P_{best} \leftarrow P_{current}$
    end while
    Return $[\mathcal{A}_S]$
end procedure

Overhead

Plat.	Target	Source	#Doc. in Target	#Doc. in Source	Time (hh:mm:ss)
IFTTT	Evernote	Coarse Location + Fine Location + Camera	145	839	33:27:03
	BMW Lab	Send SMS + Record Audio	65	296	14:08:40
	Facebook	Camera	115	11	22:57:20
	Google Calendar	Read Calendar + Coarse Location	102	207	15:15:18
	Google Contact	Read Contacts	49	17	18:40:17
Chrm.	Geolocation	Fine Location + Coarse Location + Read Contact	138	845	07:37:28
	Proxy	Send SMS + Fine Location	123	921	06:54:01
	Content Settings	Fine Location + Read Contact	58	652	09:42:45
Smart Things	Lock	Write Setting	30	31	03:47:59
	Motion Sensor	Read Contact	60	17	04:09:44
	Switch	Send SMS + Read Calendar	153	302	14:11:08

Discussion

Theory vs Practice

Evaluation

Plat.	Permission	Performance Acc.	Prec.	Rec.	F1
IFTTT	Evernote	84.6%	77.53 %	89.61%	83.13%
	BMW Lab	94.00%	99.99%	90.90%	95.24%
	Facebook	90.00%	78.72%	100%	88.09%
	G. Cal.	88.51%	86.96%	98.36%	94.30%
	G. Con.	94.11%	93.33%	100%	98.41%
Chrm.	Geoloc.	89.43%	85.96%	90.74%	88.29%
	Proxy	89.81%	89.24%	98.80%	93.78%
	C. Settings	76.74%	68.97%	95.24%	85.31%
Smart Things	Lock	93.33%	75.00%	100 %	85.71%
	Motion	82.22%	77.14%	100%	87.10%
	Switch	91.36%	89.38%	100%	94.39%

TKPERM identifies 329 overprivileged applications from all the different platforms.

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Evaluation

Plat.	Target Domain	Source Domain	Trans.	No Trans.	Improve.
IFTTT	Evernote	Coarse Location + Fine Location + Camera	83.13%	79.78%	3.35%
	BMW Lab	Send SMS + Record Audio	95.24%	85.71%	9.53%
	Facebook	Camera	88.09%	75.00%	13.09%
	Google Calendar	Read Calendar + Coarse Location	94.30%	83.54%	10.76%
	Google Contact	Read Contacts	98.41%	97.22%	1.19%
Chrome	Geolocation	Fine Location + Coarse Location + Read Contact	88.29%	62.50%	25.79%
	Proxy	Send SMS + Fine Location	93.78%	89.69%	4.09%
	Content Settings	Fine Location + Read Contact	85.31%	59.61%	25.7%
Smart Things	Lock	Write Setting	85.71%	75.00%	10.71%
	Motion Sensor	Read Contact	87.10%	53.33%	33.77%
	Switch	Send SMS + Read Calendar	94.39%	90.09%	4.3%

We find that the app overprivilege is a pervasive issues. On average, we find 32.33% of apps are overprivileged. 135 apps (28.36%) from IFTTT, 114 apps (35.73%) from Chrome Extension, and 80 apps (32.9%) from SmartThings are overprivileged.

Discussion

Did you use experimentation artifacts borrowed from the community? -- Yes our Android dataset is inherited from AutoCog, and we also publish our dataset for future research

Did you attempt to replicate or reproduce results of earlier research as part of your work? -- We try their work on different domains and didn't receive good results, which is the key motivation for this research.

What can be learned from your methodology and your experience using your methodology? -- When state-of-the-art algorithm didn't work, we can come up with better/easier solution once we understand the problem we are facing

What did you try that did not succeed before getting to the results you presented? -- We tried SDN dataset, but it doesn't include detailed description/not having enough dataset.

Next Step

- Include more target platforms such as VR/AR when they gain more popularity.
- The concept of transfer learning could also be helpful for other problems in the cybersecurity domain, for example, to analyze network traffic for different IoT platforms
- Analyze the advantage and difficulty of our transfer learning experiment in the post-workshop paper.

Thank you