2021년 한양대학교

소프트웨어융합 종합학술대회

[분야 : 산학연구개발]

제목: Tap Position Inference with Mobile Embedded Sensor Data and its Digital Twin Implementation

김은정(2019093418), 최다연(2019041167), 지홍근(2019042633)

2021. 11. 26

한양대학교 SW중심대학사업단

1. Research Introduction

What is the basis of this research?

“

I believe the Metaverse is the next chapter for the Internet.

Mark Zuckerberg, Co-founder of Meta Platforms Inc.

Metaverse and Trends

- **Metaverse**
  - A collective virtual shared space created by the convergence of virtually enhanced physical reality and physically persistent virtual space.

- **Trends**
  - To break the limits of the real world, human beings always have a huge interest in building a virtual world.
  - The necessity of advanced virtual space has been mentioned because of the online-oriented world.

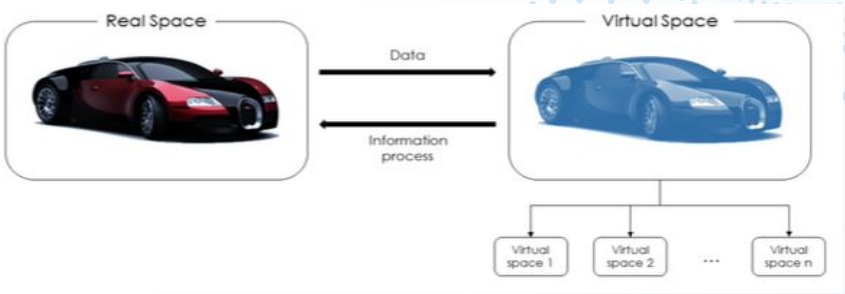
https://www.globaltimes.cn/page/202111/1238811.shtml#:~:text=The%20metaverse%20is%20a%20collective,augmented%20reality%2C%20and%20the%20internet.

The Motivation of Digital Twin

- Heavily increasing demand for the variety of products stresses the manufacturing industry to become versatile.
- Digital Twin is one of the most promising technologies for realizing smart manufacturing and Industry 4.0. (The Smart Factory)
- The seamless integration of cyber-physical and real-time monitoring has been mentioned until now.
- High fidelity and real-time communication between Physical and Virtual is required for the shop floors to accelerate the production.

The Definition of Digital Twin

A virtual representation of a physical asset enabled through data and simulators for real-time prediction, optimization, monitoring, controlling, and improved decision making.



Barricelli, Barbara Rita, Elena Casiraghi, and Daniela Fogli. "A survey on digital twin: definitions, characteristics, applications, and design implications." *IEEE access* 7 (2019): 167653-167671.

2. Tap Position Inference

How to fully understand the Physical Tapping position?

Subproject of the Digital Twin

Our Hypothesis

1. It is possible to infer the tapping position from the cheapest components of mobile devices.
2. Accelerometer, Gyroscope and GPS are mostly imbedded on mobile devices and inexpensive.
3. Those sensors must have different values depending on the user usage habits.
4. Collecting the sensor data is durable as the components exist inside the device black box.

Goal of this Subproject

1. Build an inference model to conjecture the tapping position only with the sensor data.
2. Utilize the tap position inference model to build an app usage inference.

Tapped Zone Inference Matters

Tap position Inference

Model can infer the tapping zone of the devices with the sensor data.

Privacy Problem

Conjecturing the app usage let someone backtracks the user's warm data (What it the recent interest in what?).

Reinforcement Learning

Agent can be trained by taking the sensor context information.

App Usage Inference

Inferred tapped position can lead us to infer the next phase, the app usage inference.

Security

Shut off the privacy leakage in prior whenever it is likely to inference the tap position.

i.e. App Design Shift

Digital Twin

Regard the agent as the Digital Twin of the mobile device. Utilize the digital twin as a security medium.

Digital Twin will feedback in terms of the security with a real-time and high fidelity.

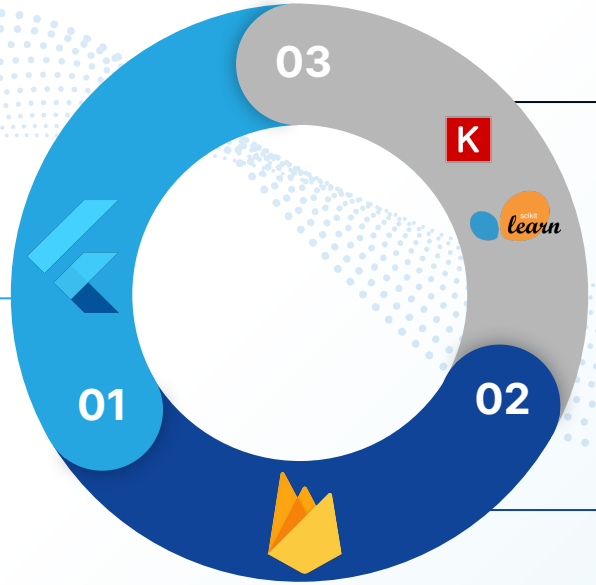
How does the Inference works?

The Big Picture

Sensor Data & Tapping Position collector App. with Flutter

Flutter App will send sensor data(Accelerometer, Gyroscope) and tapped position(x, y coordinates) to FireStore database whenever a user taps a zone.

Zones are divided into 12 parts.



Tap Position Inference Models

Input Data :
- **Three Accelerometer Data (x, y, z axis)**
- **Three Gyroscope Data (x, y, z axis)**
- **Fusion Sensor Data**

Target Data :
- **Tapped Zone**

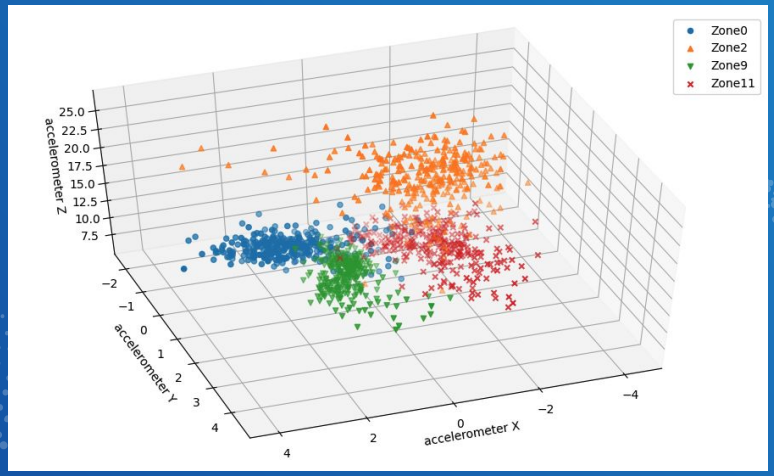
Random Forest, Decision Tree and Sequential Model with multiple layers

FireStore with NoSQL

Lightweight, Easy, and Fast Database from Google Firebase.

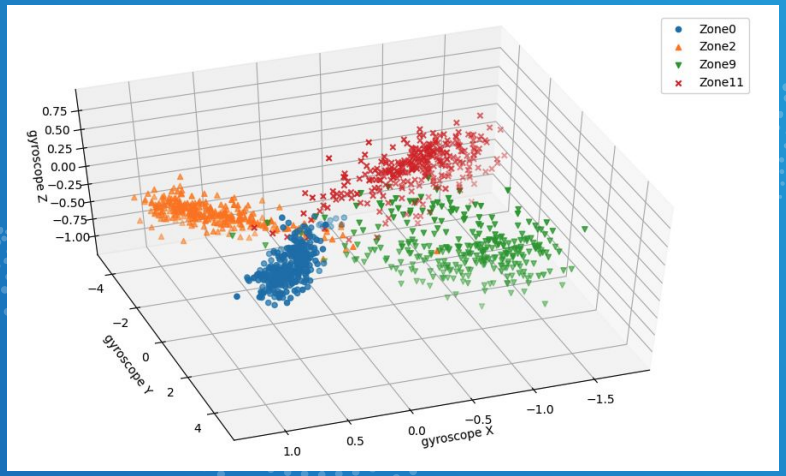
Every single document(Record) is composed of ID, sensor data and tap position.

Clusters of Sensory Data based on Tapping Zone



Different correlation of each Accelerometer values.

The values of each zone can used as a feature.

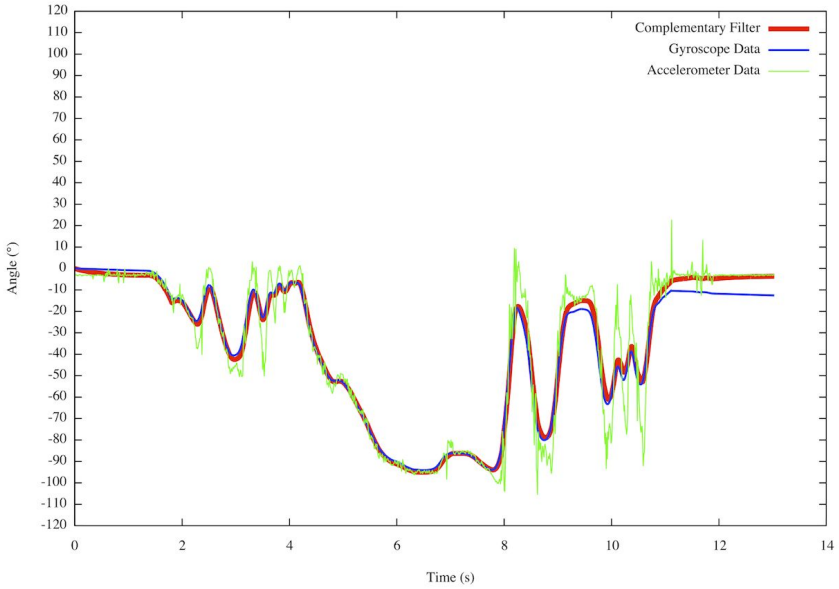


Inference can be improved with three more different Gyroscope values

Inputs and Targets

	Accelerometer_X	Accelerometer_Y	Accelerometer_Z	Gyroscope_X	Gyroscope_Y	Gyroscope_Z	Tapped_Zone (Target)
1	0.76617	0.50228	0.37037	0.23926	0.62974	0.33833	**11**
2	0.44328	0.40446	0.31342	0.56460	0.30335	0.41662	**2**
3	0.34458	0.59175	0.29805	0.30159	0.32840	0.33909	**5**
…	…	…	…	…	…	…	…
2098	0.40628	0.33470	0.60746	0.53867	0.20581	0.15687	**5**
2099	1.17715	0.18142	-0.46163	-0.07735	1.42788	0.65097	**11**
2100	-0.88317	2.18343	-1.50166	-0.01835	-1.29749	-0.03310	**0**

Sensor Data Fusion
with *Complementary Filter*



Why Sensor Data Fusion?

1. Increases the quality of the raw(noisy) data.
2. Increases reliability.
3. Helps to estimate unmeasured status.

Complementary Filter

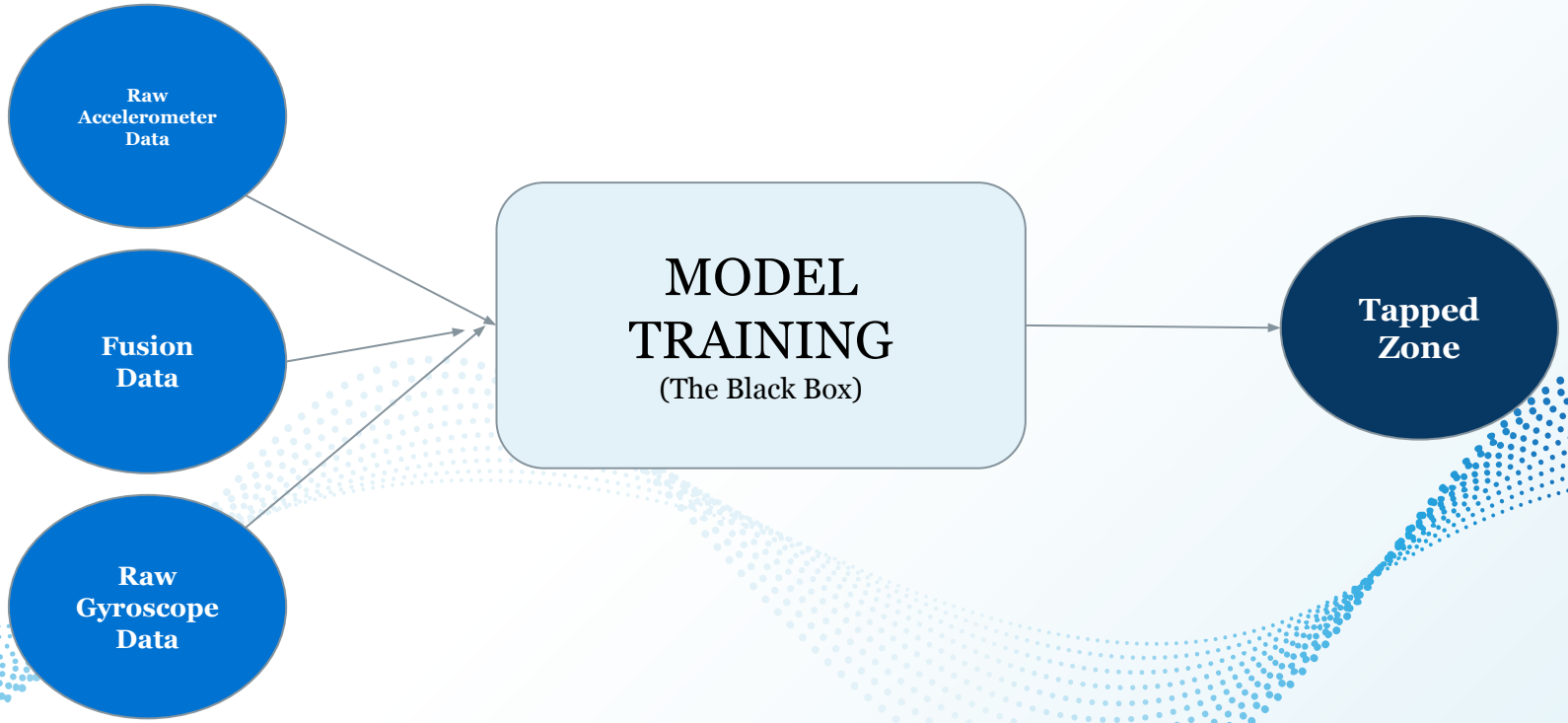
1. Simple filter that allows us to weight an incoming piece of data depending on how much we rely on.
2. For this project, we will combine Accelerometer and Gyroscope values to assist the model training.

Mathematical Expression

PITCH_C = arctan(Ay / Az)
ROLL_C = arctan(Ax / Az)
*PITCH_M = 0.98 * (PITCH_M+Gx*dt)+PITCH_C*0.02*
*ROLL_M = 0.98 * (PITCH_M+Gx*dt)+ROLL_C*0.02*

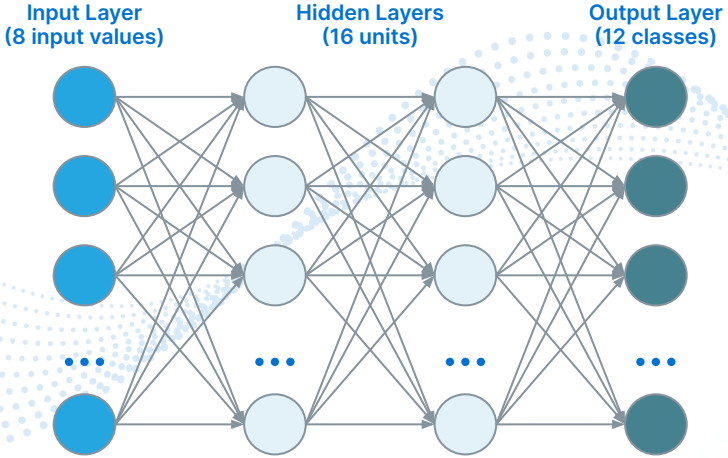
http://www.pieter-jan.com/node/11

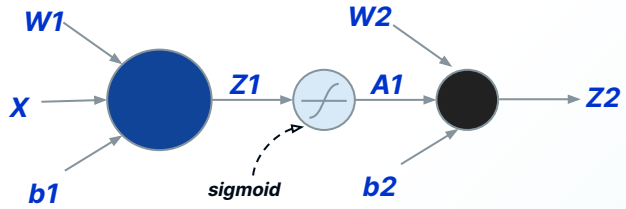
The summary of Input and Output

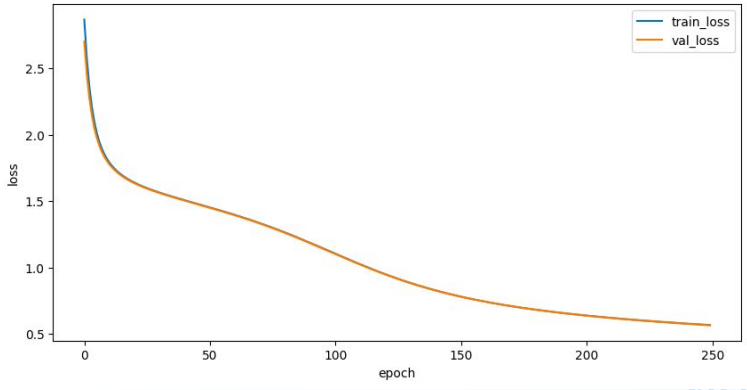


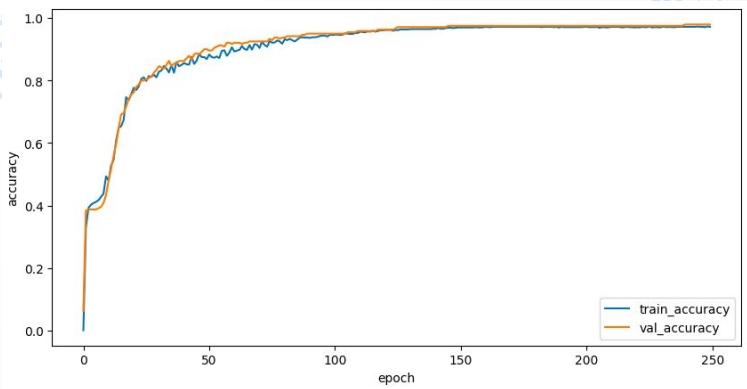
Sequential Model

with 2-hidden Layers









97% Performance for Testing

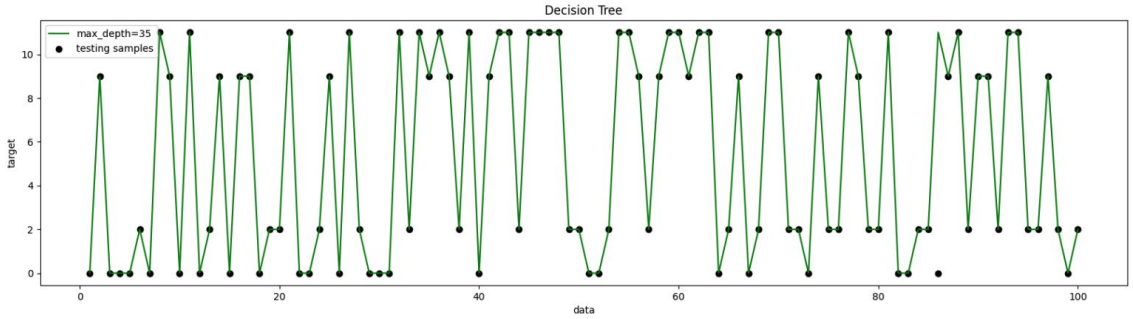
88% Performance for Practice

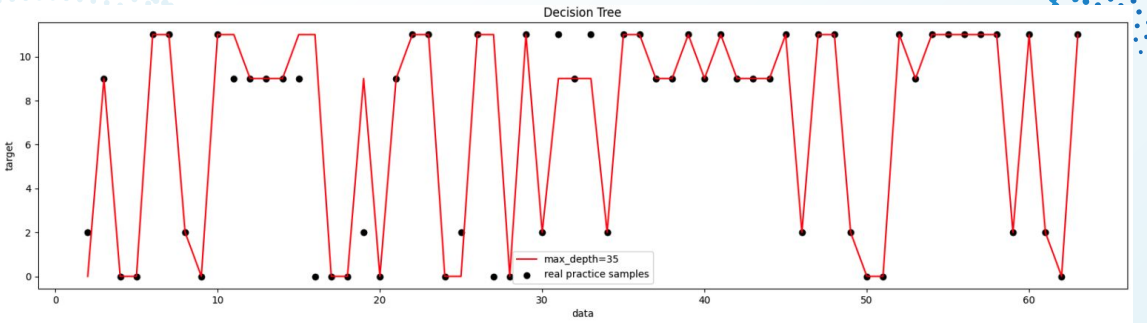
Decision Tree Model

with maximum 35 depths

95% Performance for Testing

82% Performance for Practice



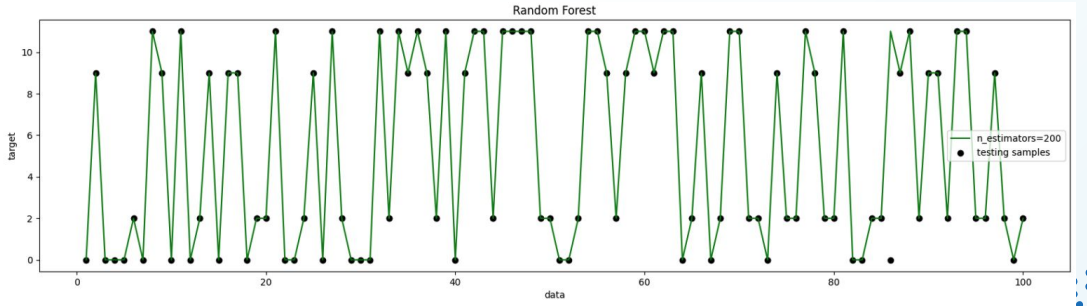


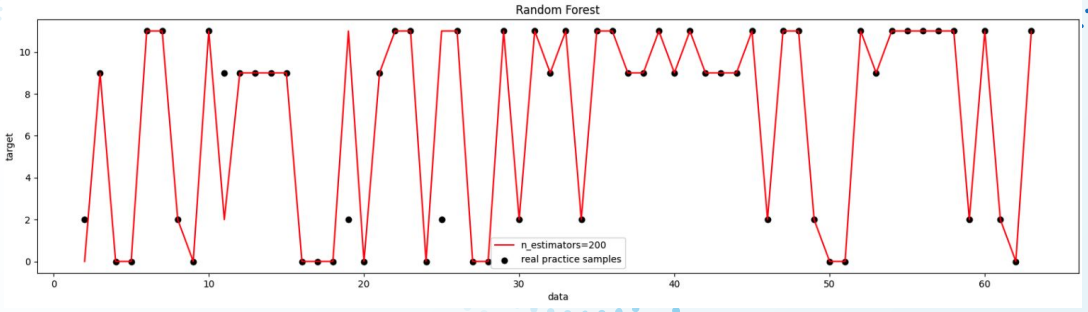
Random Forest Model

with maximum 35 depths, n_estimators = 100

98% Performance for Testing

88~89% Performance for Practice

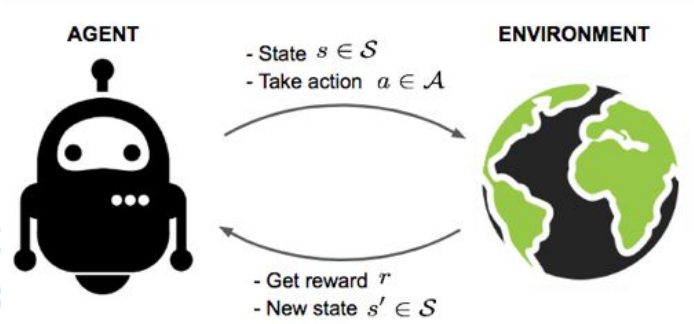




3. Implementation of Digital Twin

Our Next Step…

Digital Twin with Reinforcement Learning



Digital Twin Agent keep resides on the background.

The Agent tries to figure out the situation on the mobile usage with sensor data.

Environment *:= Upcoming unpredictable Sensor Data*

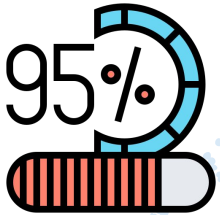
State *:= Set of the User's Device*
(i.e. Sensor status, Fusion Data status, and App Usage Status)

Action *:= Depends on what the Digital Twin does*
(i.e. Pre-Load, Context pre-fetch, and App status Change)

Reward *:= Depends on what the Digital Twin does*
(i.e. Pre loading correction)

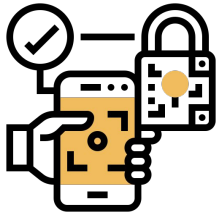
https://lilianweng.github.io/lil-log/2018/02/19/a-long-peek-into-reinforcement-learning.html

Digital Twin with Reinforcement Learning (cont.)



Pre-Loading and Context Pre-Fetching

Trained Digital Twin Agent with Sensor Data can interact with mobile devices in terms of loading or context switching. Inheriting the properties of Legacy Digital Twin, the Agent assists the loading session in a real-time, control with high-fidelity from seamless integration of mobile devices.



App. environment variation

Due to the possibility of app usage inference, the necessity of mobile app usage protection has emerged. Trained Digital Twin Agent can shut off the invalid privacy approach by intentionally change the App. environment. This happens when the Agent thinks personal information leakage is suspected.

The way how to protect the private information includes, app design change or mobile usage environment change.

Zhihao Sehn et al. "DeepApp: A Deep Reinforcement Learning Framework for Mobile Application Usage Prediction" *IEEE* (2021): 1-1.

Conclusion

1. **Digital Twin has emerged.**
   Due to the human desire to explore the Virtual World, the Digital Twin concept was born, and it is reasonable to study this concept in universities and research institutes.

2. **Sensory Data can be a cornerstone of Mobile Environment Understanding.**
   Building a tap position inference model is possible when sufficient data are given.

3. **Tap Position Inference can lead the Mobile Device usage Inference.**
   From the inference model, we can also infer the personal information and app usage of the device. Privacy leakage is possible because of this scenario.

4. **Digital Twin developed by Reinforcement Learning has advantages.**
   As we discussed, it is valid to use the Machine Learning technique for an inference using reinforcement learning to build a Digital Twin Agent is encouraged, and this can bring positive effects such as pre-loading and privacy protection.

Here is the project implementation with Python and Dart -> ***https://github.com/pithecuse527/Tap-Position-Inference***

Thank you!

Will be back soon with the Advanced Digital Twin...