

Reproducibility in Computational Research

“ A Predictive DASH QoE Approach Based on Machine Learning at Multi-access Edge Computing ”

Md Tariqul Islam

MSc Student

FEEC, UNICAMP

mtarislam@gmail.com



Agenda

- What is Reproducible Research
- Importance of Reproducibility
- How to Achieve Reproducibility
- Project Overview
- Project Reproducibility
- Challenges and Lessons



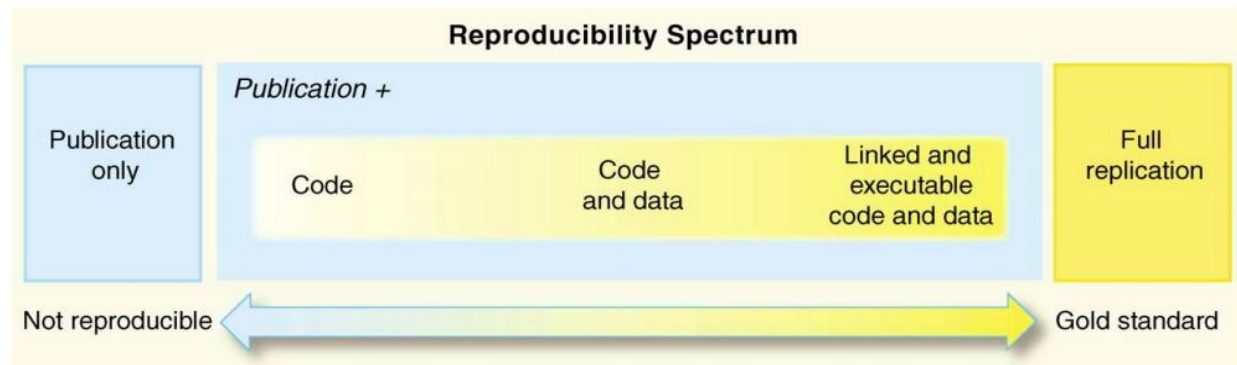
What is Reproducible Research?





“Again, and Again, and Again ...” BR Jasny et. al. Science, 2011.
334(6060) pp. 1225 DOI: 10.1126/science.334.6060.1225

“Reproducible Research in Computational Science”. RD Peng Science,
2011. 334 (6060) pp. 1226-1227 DOI: 10.1126/science.1213847

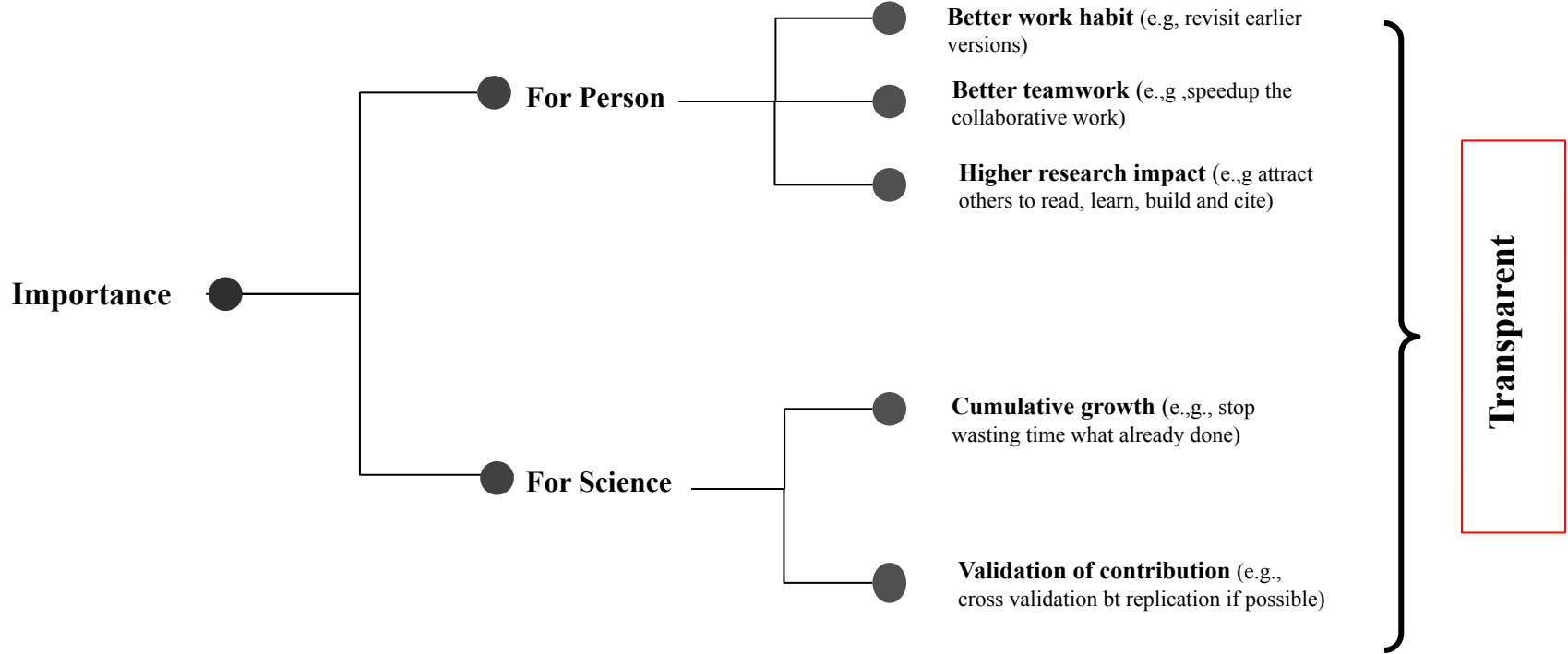


Reproducibility in Computational Research is an exercise to make available of all data, code, and required tools for others to reproduce the same results discussed in original research work.



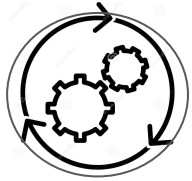
Importance of Reproducibility





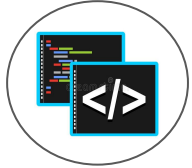
How to Achieve Reproducibility?





Workflow

- One knows exactly what path work should take.
- Well, defined inputs and outputs joined in a pipeline.
- Automate the pipeline as much as possible



Code

- Good coding structure.
- Keep track record on changes
- Keep record of random seeds



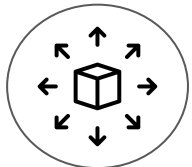
Data

- Auto data manipulation for easy to re-use
- Treat metadata as data besides raw and preprocessed data
- Keep track history of data provenance to defend conclusion
- What data should store and share



Documentation

- Documentation on data generate, process and analyze
- Documentation on code-purpose of each section of code
- Documentation on experiment-how to execute the work

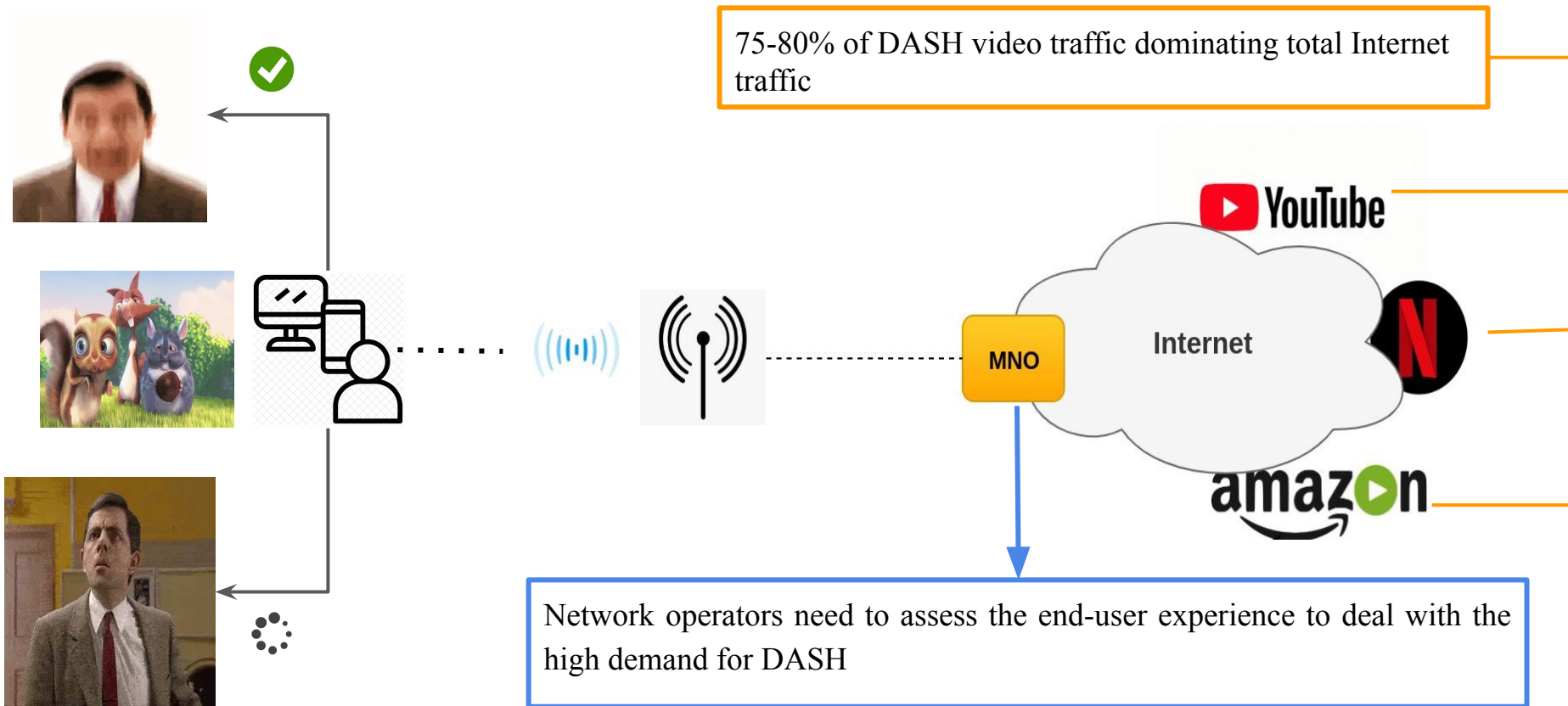


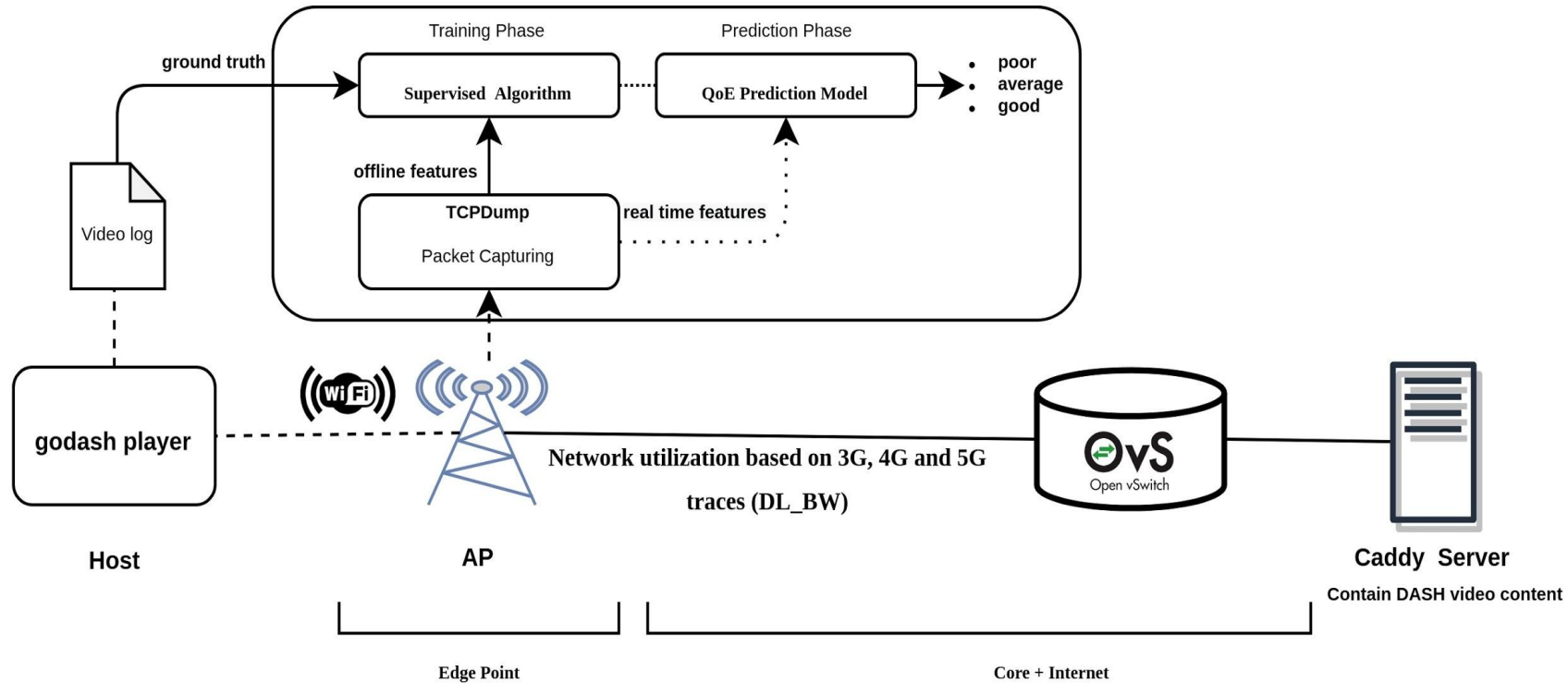
Distribution

- Give public access of code, data
- Archive and share all dependencies, libraries and tools with exact version
- Share the computing environment in container, virtual machine or cloud host

Project: *A Predictive DASH QoE
Approach Based on Machine
Learning at Multi-access Edge
Computing*

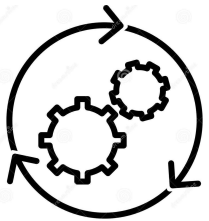






Project Reproducibility



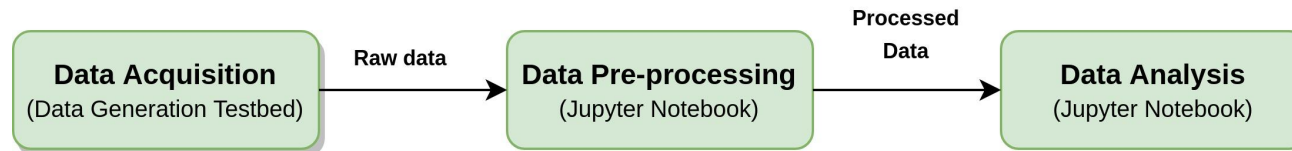


Workflow

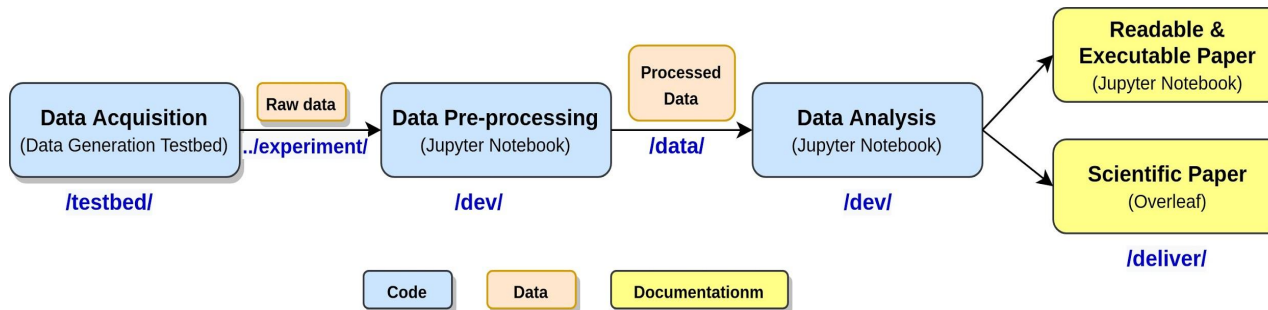
Initial

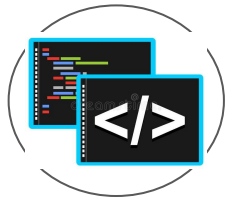


Intermediary

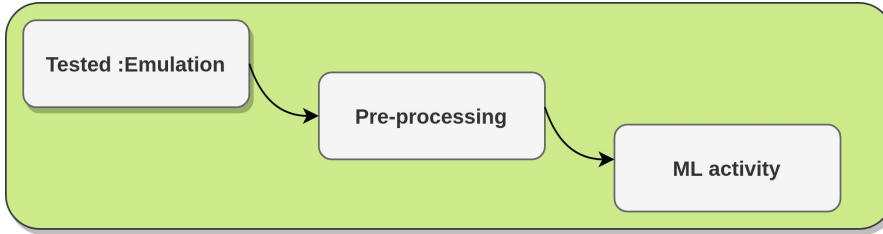


Final





Code



Literate Programming =
human readable (text) +
machine readable (code)



```
commit b6ba7e55bae0c47965950c5476b453aa58d9/ca5 (HEAD -> master, origin/master, origin/HEAD)
Author: sajbibtariq <sajib.tariq12@gmail.com>
Date: Sun Jun 28 04:27:23 2020 -0300

    minor changes

commit b2e174ae28b6c797ca50281f1659dd4ce5c5f8e2
Author: sajbibtariq <sajib.tariq12@gmail.com>
Date: Sun Jun 28 04:25:18 2020 -0300

    minor changes
```

Track over changes =
Version control (git) + store
repository (github)



```
= train_test_split( X, y, test_size=0.1, random_state=42)

target = smape
smote=SMOTE(random_state=42) # resample all
X, y = smote.fit_resample(feature_target)
```

Control randomness =
Seeds (fixed)





Data

Auto Data Manipulation Script
data preprocess, plotting graph

Treat Metadata as data
input attributes of raw data
generation and preprocessed
dataframe header

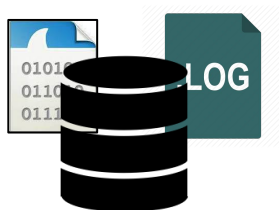
Keep a copy of raw data
network traffic and video log for
transparent on data provenance



```
mode=['3g','4g','5g'] #network type '5g',
host=[1] # number of host
algo=['conv'] # adaptation algorithm 'conventional'
net3= ['metro','bus', 'train', 'ferry','car'] # mobility for 3g
net4= ['bus', 'train', 'static','car','pedestrian'] # mobility for 4g
net5= ['A_A_Static','D_Driving', 'D_Static'] # 'A_A_Static','A_A_Driving
doc3=['Am'] # number of operator (1)
doc4=['Am','Bm'] # number of operator (2)
doc5=['Bm'] # number of operator (1)
num = [1,2,3] # number of traces
```

Entire dataset including all the metrics

	Type	Mobility	Operator	Trace	Total host	Client	Algorithm	Port	Segment	\
0	3g	metro	Am	1.0	1	1	conv	58428	1	
1	3g	metro	Am	1.0	1	1	conv	58432	2	
2	3g	metro	Am	1.0	1	1	conv	58436	3	





Documentation

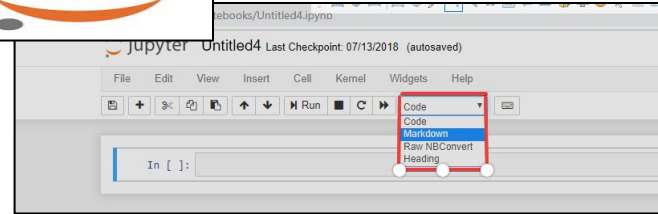
Readable & Executable Paper
Jupyter notebook using Markdown
feature

Scientific Paper
PDF format

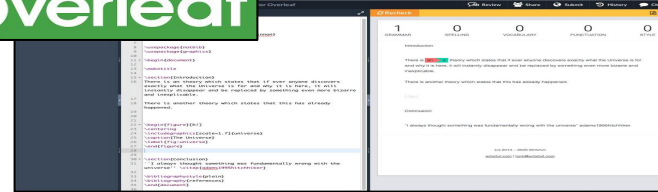
Github Readme
workflow, requirements, folder
structure scheme, and how to use
codes and data



NOTEBOOK



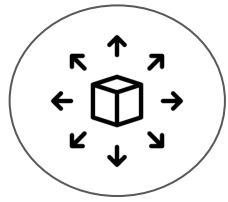
PDF



GitHub

README



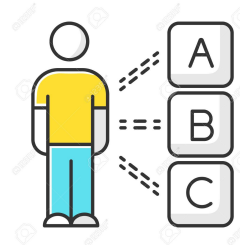


Distribution

**Public access to code
and data**



**Provide multiple options to
reproduce the work in the
shared project**



**Computational environment
wrapped with everything**



- ✓ Code
- ✓ Data
- ✓ Workflow
- ✓ Documentation
- ✓ Distribution

Embrace Reproducibility All Keys

Project Shared on Github

Challenges





To fulfill every reproducibility criteria



Make an understandable document for others



Wrapping the computational environment in virtual machine



Store large data set

Lessons



Current:

- ✓ Maintain documentation (code and data)
- ✓ Version control (control)
- ✓ Archiving all dependencies with the exact version

Future:

- Docker to wrapping all packages as a lightweight container
- Data version control and sharing (e.g., zenodo, kaggle)



Thanks!

Questions?

