Addressing Verification and Validation Challenges in Future Cyber-Physical Systems

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Sklodowska-Curie grant agreement No 823788.

On Deploying Machine Learners into Embedded Systems

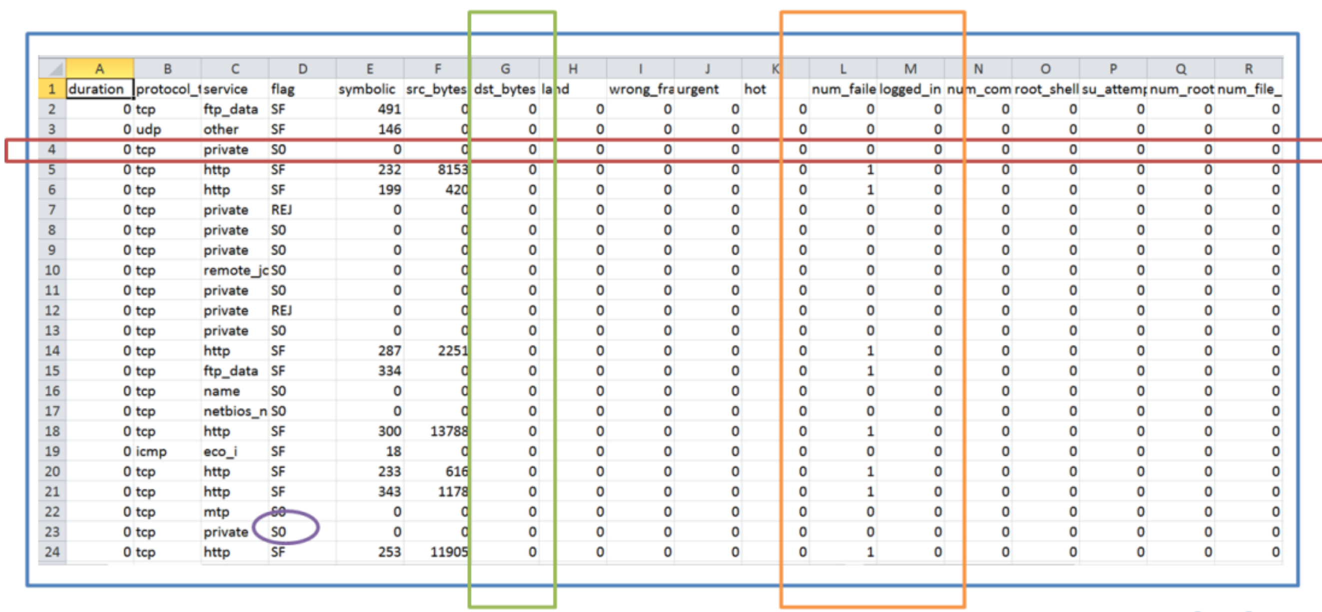
Tommaso Zoppi, Andrea Ceccarelli, Andrea Bondavalli
RCL Group – University of Florence - Italy
e-mail: tommaso.zoppi@unifi.it

RCL
RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI
DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

Tabular Data

- Embedded and General-Purpose systems often share the need of analysing tabular data
  - Features: system indicators (mainly networks)
  - Label: normal behavior or specific type of attack



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	duration	protocol_t	service	flag	symbolic	src_bytes	dst_bytes	land		wrong_fra	urgent	hot	num_faile	logged_in	num_com	root_shell	su_attemp	num_root	num_file_
2	0	tcp	ftp_data	SF	491	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	udp	other	SF	146	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	tcp	http	SF	232	8153	0	0	0	0	0	1	0	0	0	0	0	0
6	0	tcp	http	SF	199	420	0	0	0	0	0	1	0	0	0	0	0	0
7	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	tcp	remote_jc	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	tcp	private	REJ	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	tcp	http	SF	287	2251	0	0	0	0	0	1	0	0	0	0	0	0
15	0	tcp	ftp_data	SF	334	0	0	0	0	0	0	1	0	0	0	0	0	0
16	0	tcp	name	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	tcp	netbios_n	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	tcp	http	SF	300	13788	0	0	0	0	0	1	0	0	0	0	0	0
19	0	icmp	eco_i	SF	18	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	tcp	http	SF	233	616	0	0	0	0	0	1	0	0	0	0	0	0
21	0	tcp	http	SF	343	1178	0	0	0	0	0	1	0	0	0	0	0	0
22	0	tcp	mtp	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	tcp	private	S0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	tcp	http	SF	253	11905	0	0	0	0	0	1	0	0	0	0	0	0

What Anomalies are?

Anomaly detection refers to the problem of finding patterns in data that do not conform to an expected behaviour[1]

[1] Chandola, Varun, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." ACM computing surveys (CSUR) 41.3 (2009): 15.

Purpose of Anomaly Detectors

- Anomalies may have many root causes
  - Security threats
  - Misconfigurations
  - Performance Issues
  - Wrong/Slow interactions with other devices
  - Benign alterations

- Regardless of their root cause, it is always beneficial to detect them.

Embedding Anomaly Detectors

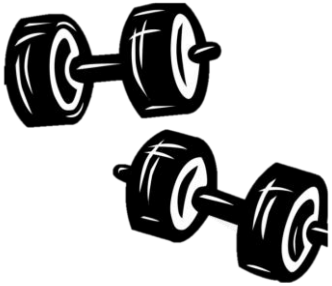
- Anomaly detectors usually rely on supervised/unsupervised ML algorithms
  - Which are usually resource and time-consuming
  - Not a huge problem for systems that do not have hardware or real-time constraints

BUT BUT BUT

- There always exists some kind of limitation to develop systems "in practise"
  - Thus, assuming "unlimited resources" is not doable

Then what?

- As a result, the best intrusion/error/anomaly detector or failure predictor for a given system must be chosen according to constraints:
  - Model size
  - Model speed
  - Detection performance
  - Availability of labelled data for training
  - …

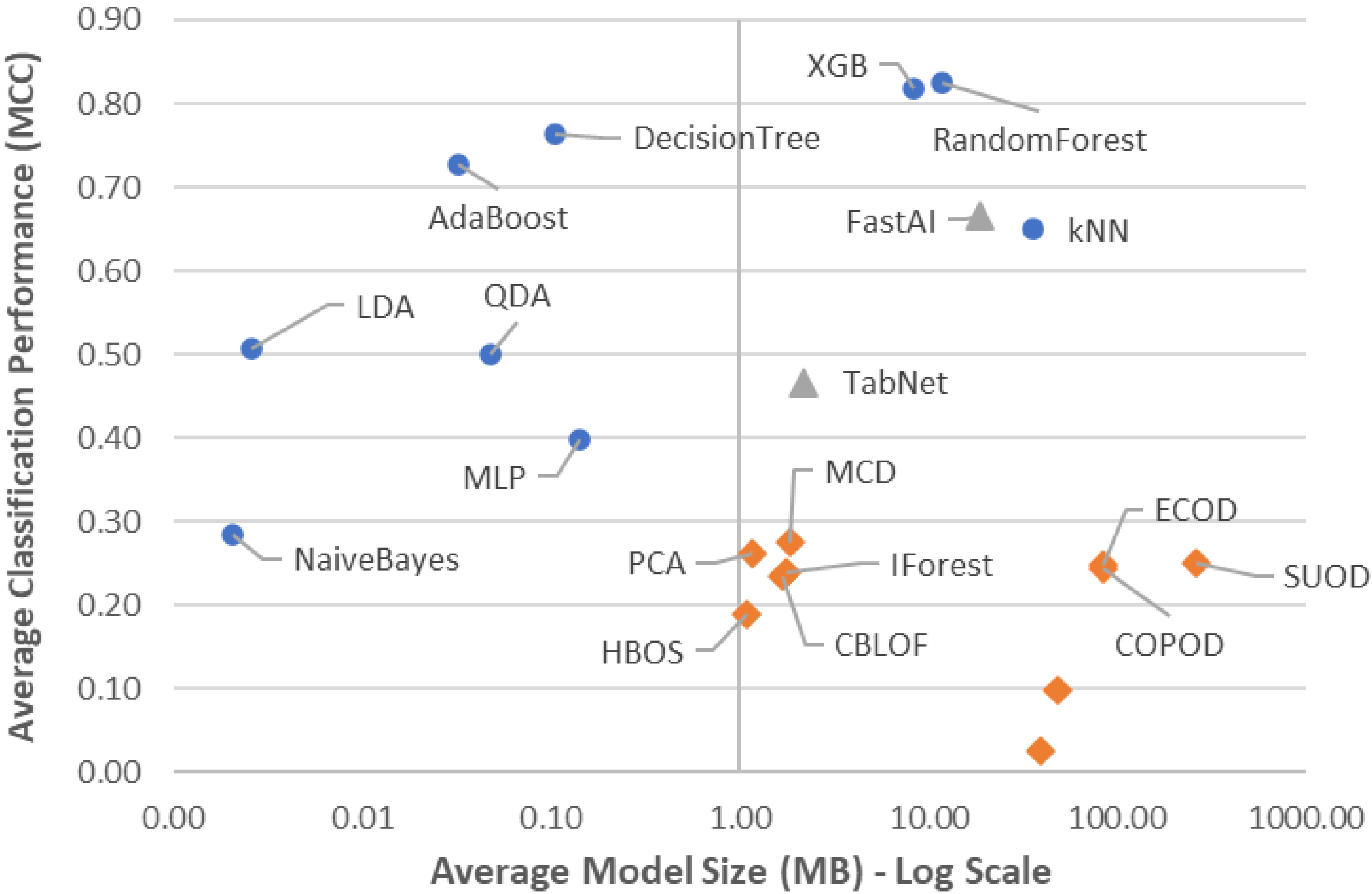


Snapshot of ML for tabular data

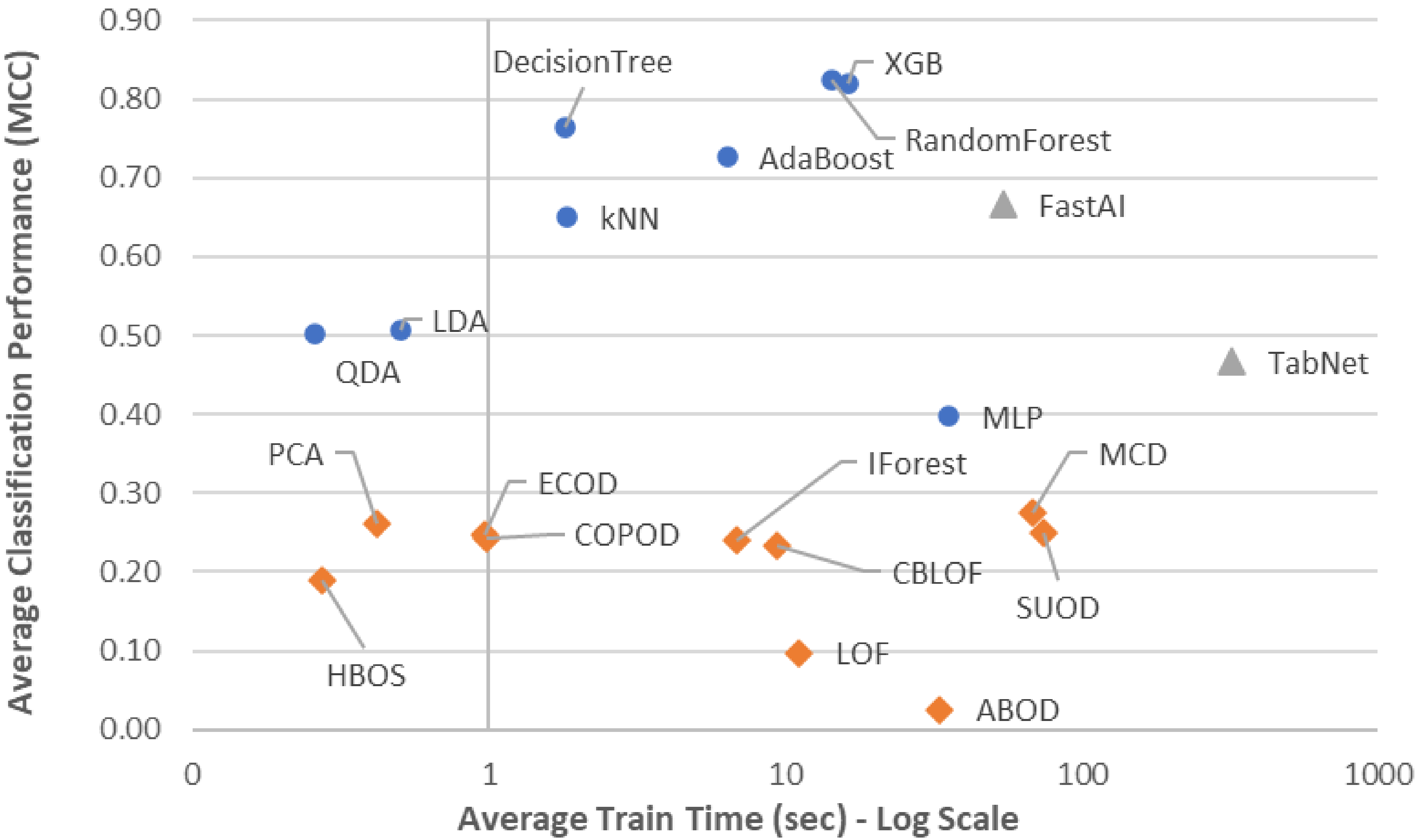
- That is why we took several SotA ML algorithms
  - Supervised: DecisionTree, RandomForest, XGB, NaiveBayes, LDA, kNN, MLP, AdaBoost, QDA
  - Unsupervised: COPOD, ABOD, HBOS, MCD, PCA, ECOD, LOF, CBLOF, Iforest, SUOD
  - Deep learning: TabNet, FastAI

- And we exercised them on a total of 33 datasets regarding critical systems to derive their average performance metrics

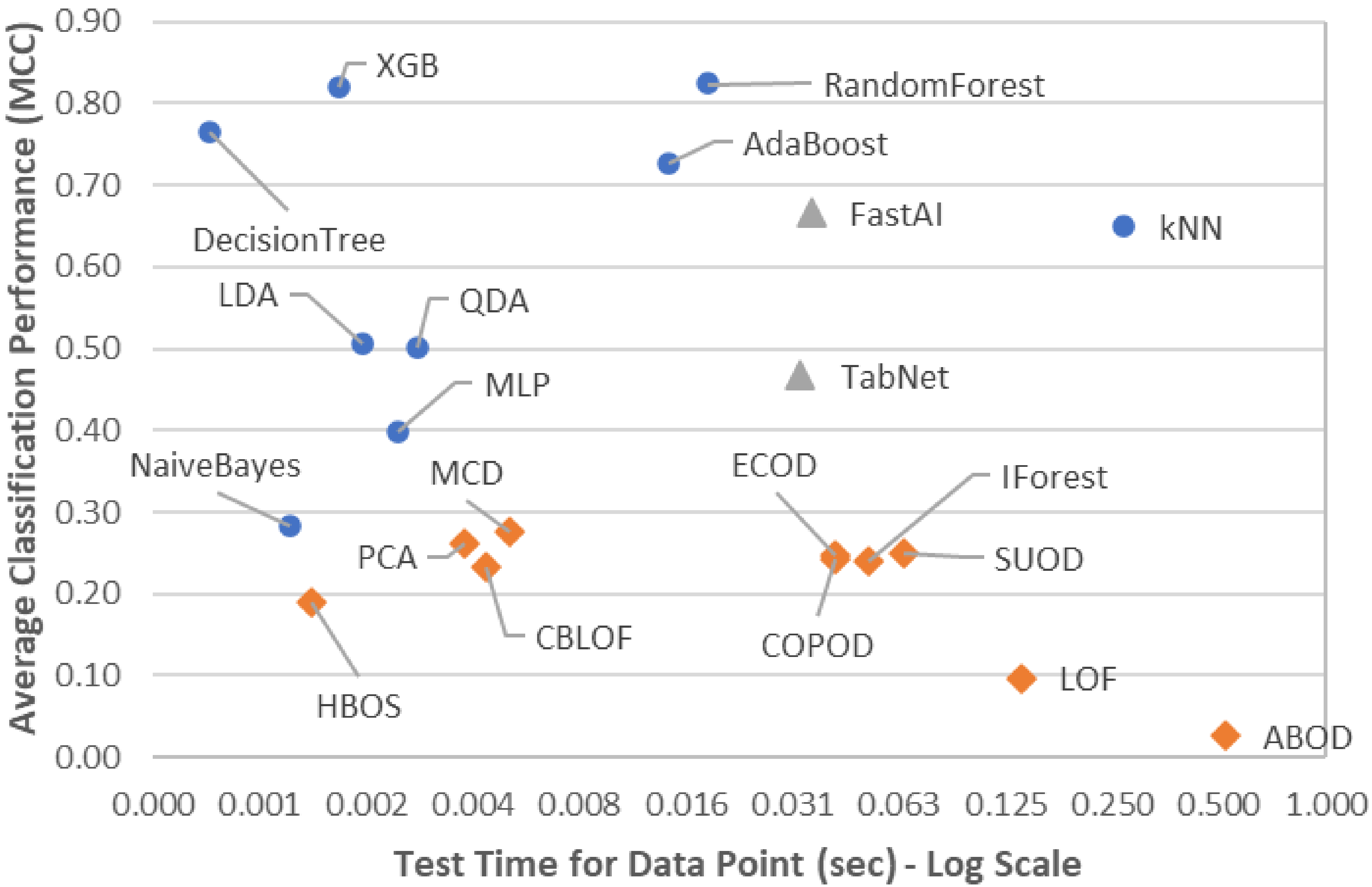
Model Size of Detectors



Model Speed of Detectors (I)

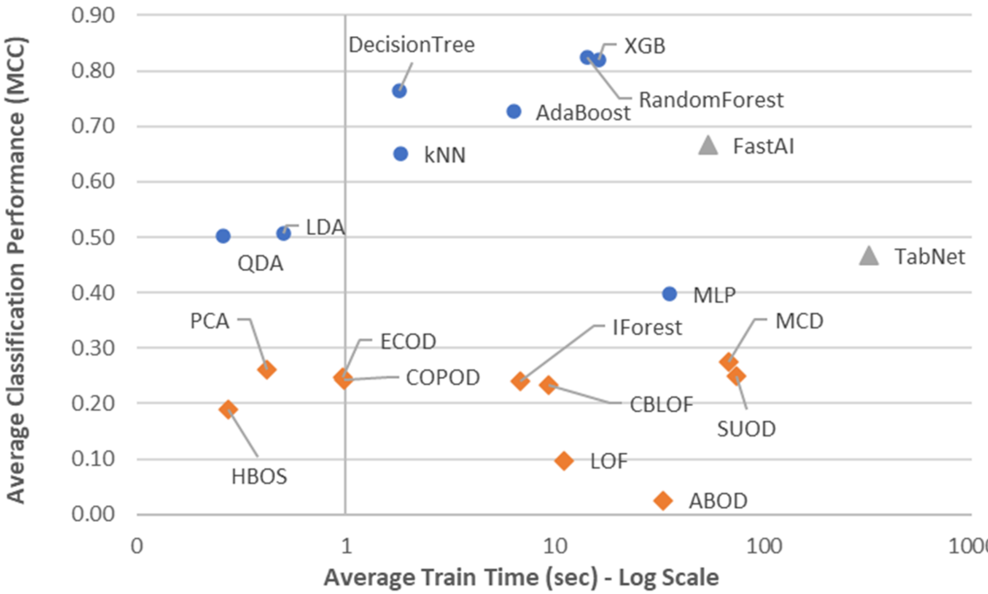


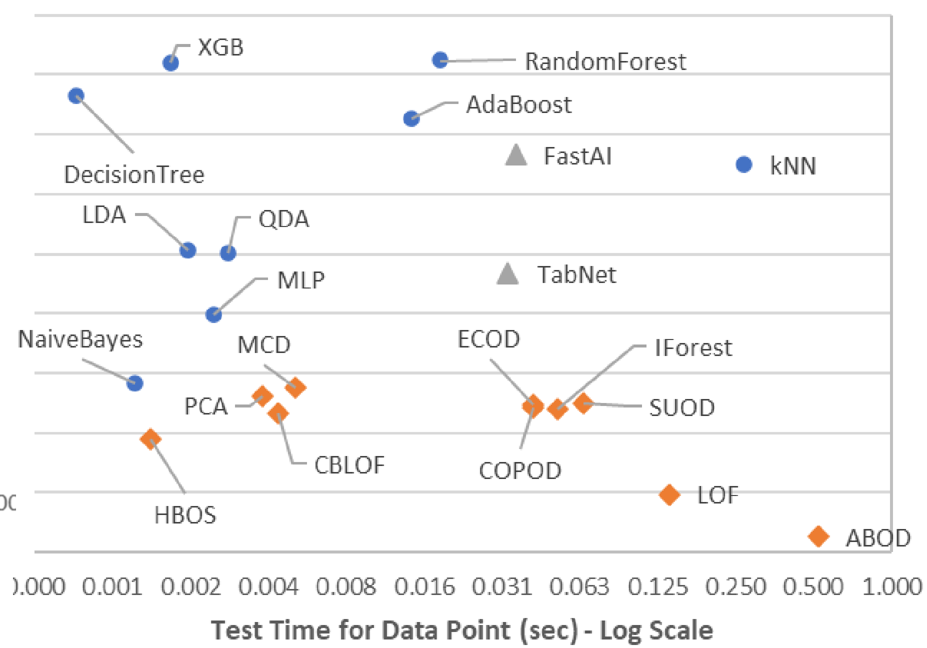
Model Speed of Detectors (II)



Model Speed of Detectors - Comments

- There are fast and slow algorithms
  - But also there are some that are fast during training and slow at runtime e.g., neighbour-based ones
  - and vice versa





Beware!

- Those numbers partially confirm the rather recent works stating that

We should not think about deep learning as the panacea for any classification task!

- For tabular data, tree-based classifiers are more interpretable, often faster and output fewer misclassifications
  - Good news for devices with limited resources!
  - See: Shwartz-Ziv, Ravid, and Amitai Armon. "**Tabular data: Deep learning is not all you need**." Information Fusion 81 (2022): 84-90 (from AI ML group at Intel Israel)

RESILIENT COMPUTING LAB

UNIVERSITÀ DEGLI STUDI FIRENZE
DIMAI DIPARTIMENTO DI MATEMATICA E INFORMATICA "ULISSE DINI"

(Finally!) Wrapping Up…

- This talk went through common constraints in deploying ML into embedded systems
  - There is no "silver bullet" algorithm to plug into a system for excellent detection capabilities and performance
  - Detectors have to be crafted for specific systems depending on their constraints
  - Availability of labels for training data is always scarce
  - This calls for unsupervised detectors, which usually have poor detection capabilities
    - There are (research) works in the direction of making unsupervised ML more accurate
    - Get in touch with us if interested!

Q&A Time


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
TASHAKKUR ATU
SUKSAMA
GRAZIE
MEHRBANI
BOLZÏN
MERCI
JUSPAXAR
SHUKRIA


Q
&
A