Machine Learning
Deep Learning
and
Knowledge Engineering

Giang Nguyen

IISAS - 20180326

ML/DL from the practical viewpoint

Development

- Strange box with math and theory: big buzz words, complicated explanations
- Can require computation power
- Hyper-parameter turning alias parametric study
  - Embarrassingly i.e. perfect parallelism for data parallelism without MPI

Production

- Can be fast + requires less computation power
- One (or a few) concreate method(s) with a little setting
  - → Which method is suitable for the given problem?
  - → Reimplementation of the selected method OR use the same tool as in development?
- Some data transformation is needed, no problem at the first view

Data models

- Build **DATA MODELS** and to use them in production
- A quest (like a math work) with more possible solutions
  - Raw data → (How? How many ways?) → Input data for ML
- ML/DL people … trying the best
  - in the time-constrain given to them
  - based on their experiences and knowledge
    - in ML/DL and
    - in the domain
- Some datasets do not have adequate quality to create useable models

ML data format

The training inputs

$\mathbf{x} = [x_1, x_2, \ldots, x_D]$ input vector $\mathbf{x} \in R^D$

$\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$ input matrix $\mathbf{X} \in R^{N \times D}$

R set of real numbers of size(s)

The targets

$\mathbf{t} = [t_1, t_2, \ldots, t_C]$ target vector $\mathbf{t} \in R^C$

$\mathbf{T} = \{\mathbf{t_1}, \mathbf{t_2}, \ldots, \mathbf{t_N}\}$ target matrix $\mathbf{T} \in R^{N \times C}$

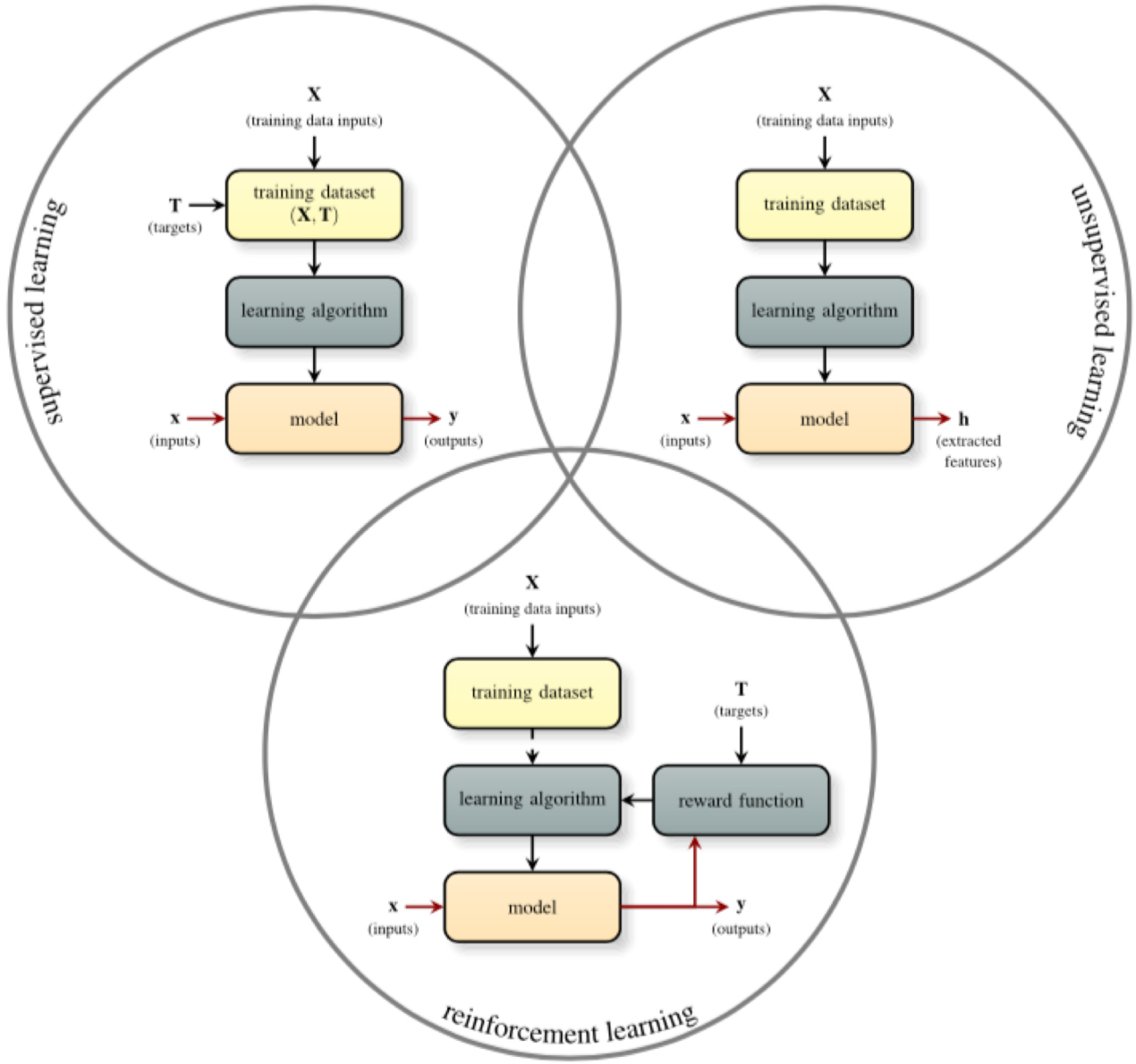
The mapping function

f: $R^D \rightarrow R^C$

The outputs

$\mathbf{y} = [y_1, y_2, \ldots, y_C]$ target vector $\mathbf{y} \in R^C$

$\mathbf{Y} = \{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_N}\}$ target matrix $\mathbf{Y} \in R^{N \times C}$



Lopes, N. and Ribeiro, B., 2015. *Machine Learning for Adaptive Many-Core Machines - A Practical Approach*. Studies in Big Data 7, DOI: 10.1007/978-3-319-06938-8_4, Springer

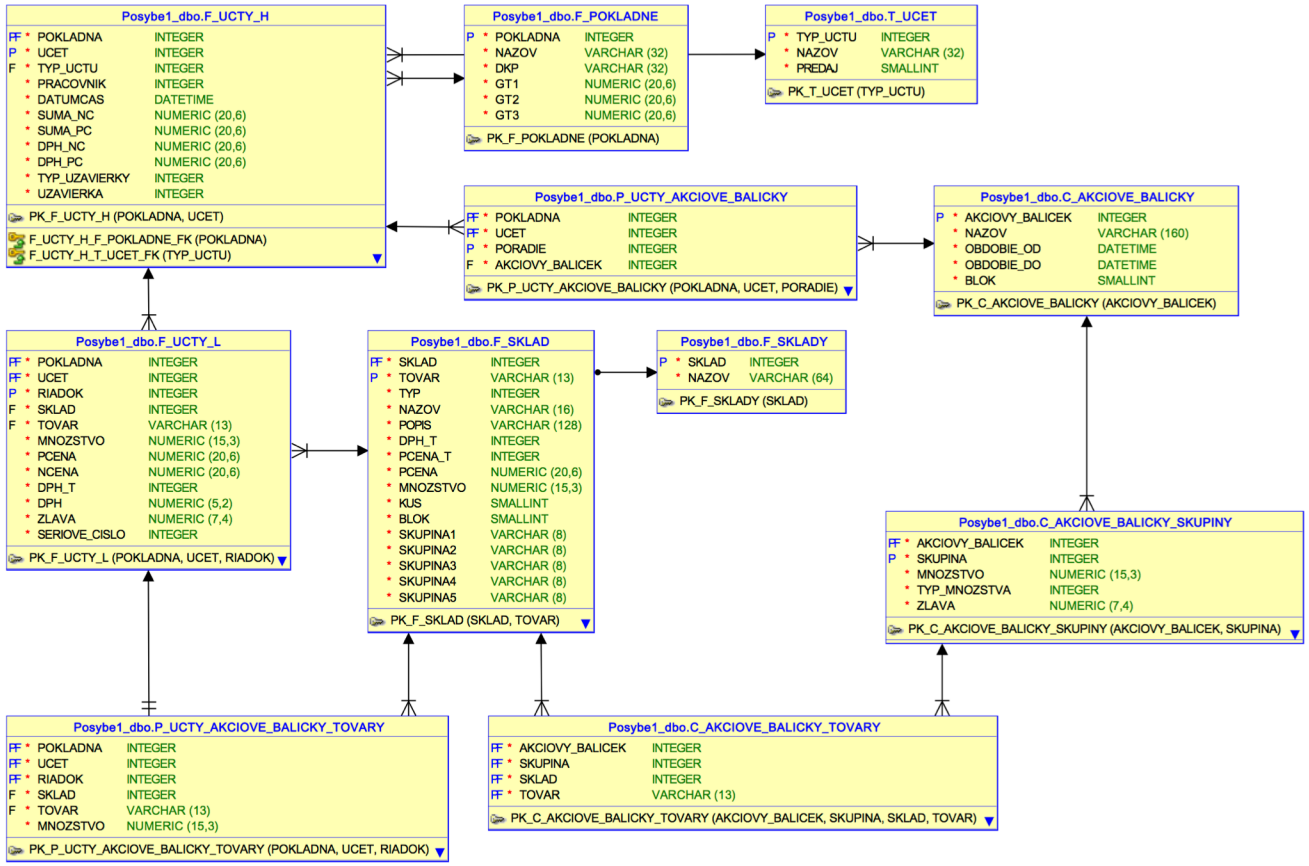
Raw data – various formats

8.4. Process log structure

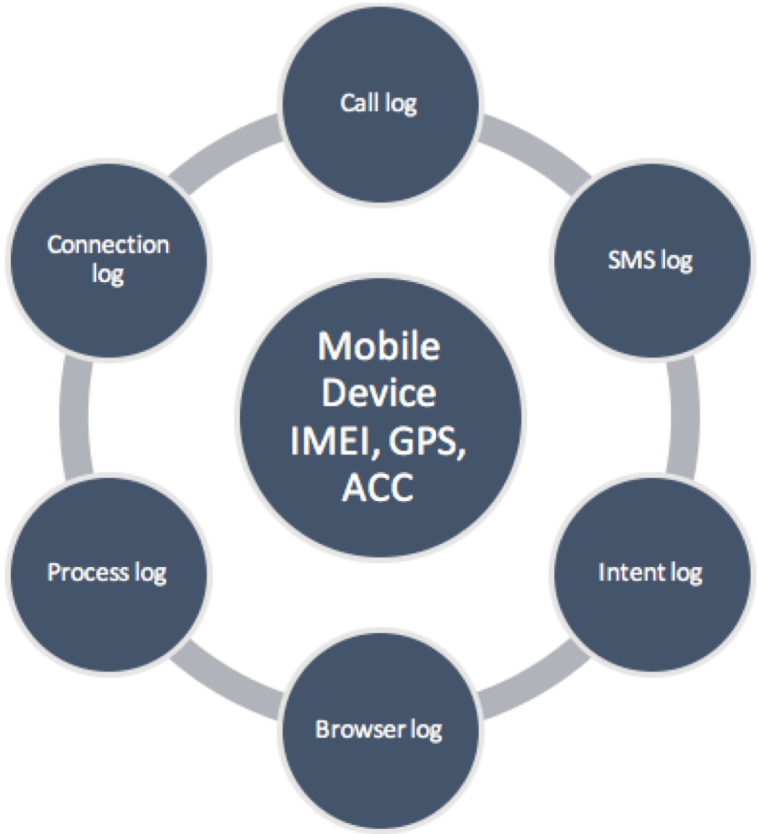
Timestamp TIMES TAMP , Processname VARCHAR2 (100) , CPUUsage VARCHAR2 (20) , LRU NUMBER (7) , ImportanceReasonPID NUMBER (7) , Importance NUMBER (7) , ImportanceReasonCode NUMBER (7) , dalvikPrivateDirty VARCHAR2 (20) , dalvikSharedDirty VARCHAR2 (20) , dalvikPss VARCHAR2 (20) , nativePrivateDirty VARCHAR2 (20) , nativeSharedDirty VARCHAR2 (20) , nativePss VARCHAR2 (20) , otherPrivateDirty VARCHAR2 (20) , otherSharedDirty VARCHAR2 (20) , otherPss VARCHAR2 (20) , TotalPrivateDirty VARCHAR2 (20) , TotalSharedDirty VARCHAR2 (20) , LongLat VARCHAR2 (100) , Longitude NUMBER (16,14) , Latitude NUMBER (16,14) , AccXYZ VARCHAR2 (100) , AccX NUMBER (12,6) , AccY NUMBER (12,6) , AccZ NUMBER (12,6) , RowID NUMBER (20) , IMEI VARCHAR2 (16) , UserID VARCHAR2 (6) , PID NUMBER (15)

8.5. Connection log structure

Timestamp TIMESTAMP , Application VARCHAR2 (100) , ToADDR VARCHAR2 (100) , ToPort NUMBER (5) , FromADDR VARCHAR2 (100) , FromPort NUMBER (5) , State VARCHAR2 (2) , LongLat VARCHAR2 (100) , Longitude NUMBER (16,14) , Latitude NUMBER (16,14) , AccXYZ VARCHAR2 (100) , AccX NUMBER (12,6) , AccY NUMBER (12,6) , AccZ NUMBER (12,6) , RowID NUMBER (20) , IMEI VARCHAR2 (16) , UserID VARCHAR2 (6)



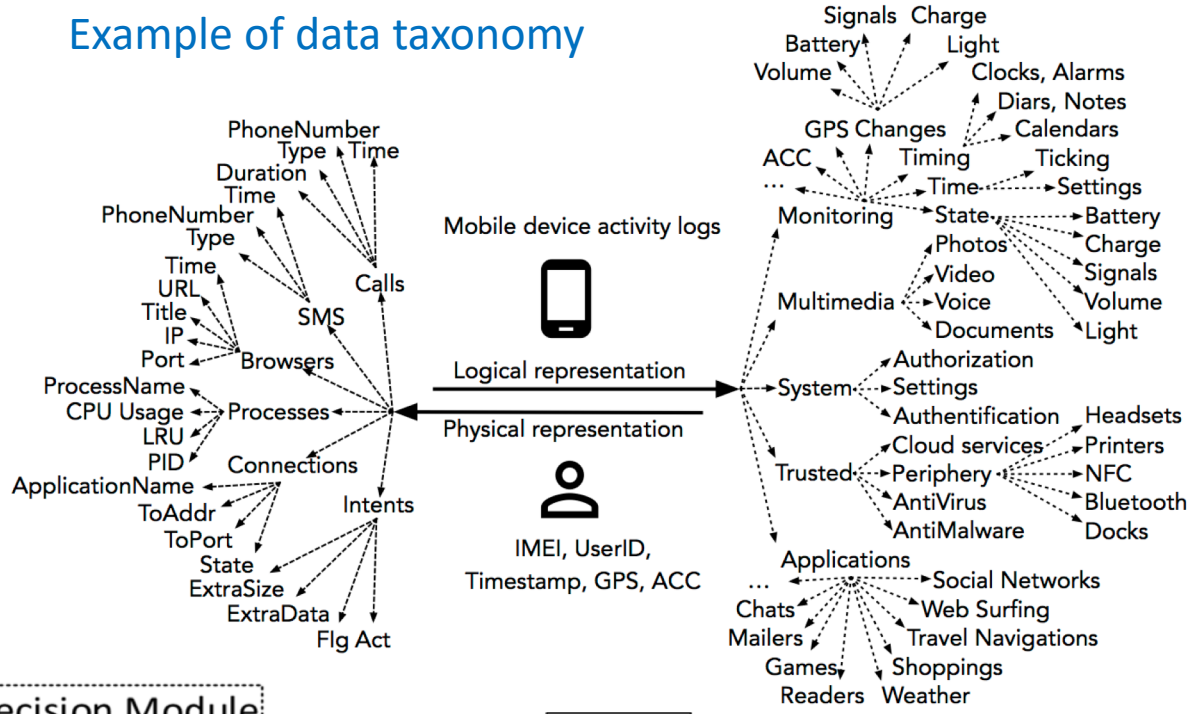
Exploratory Data Analysis (EDA example)

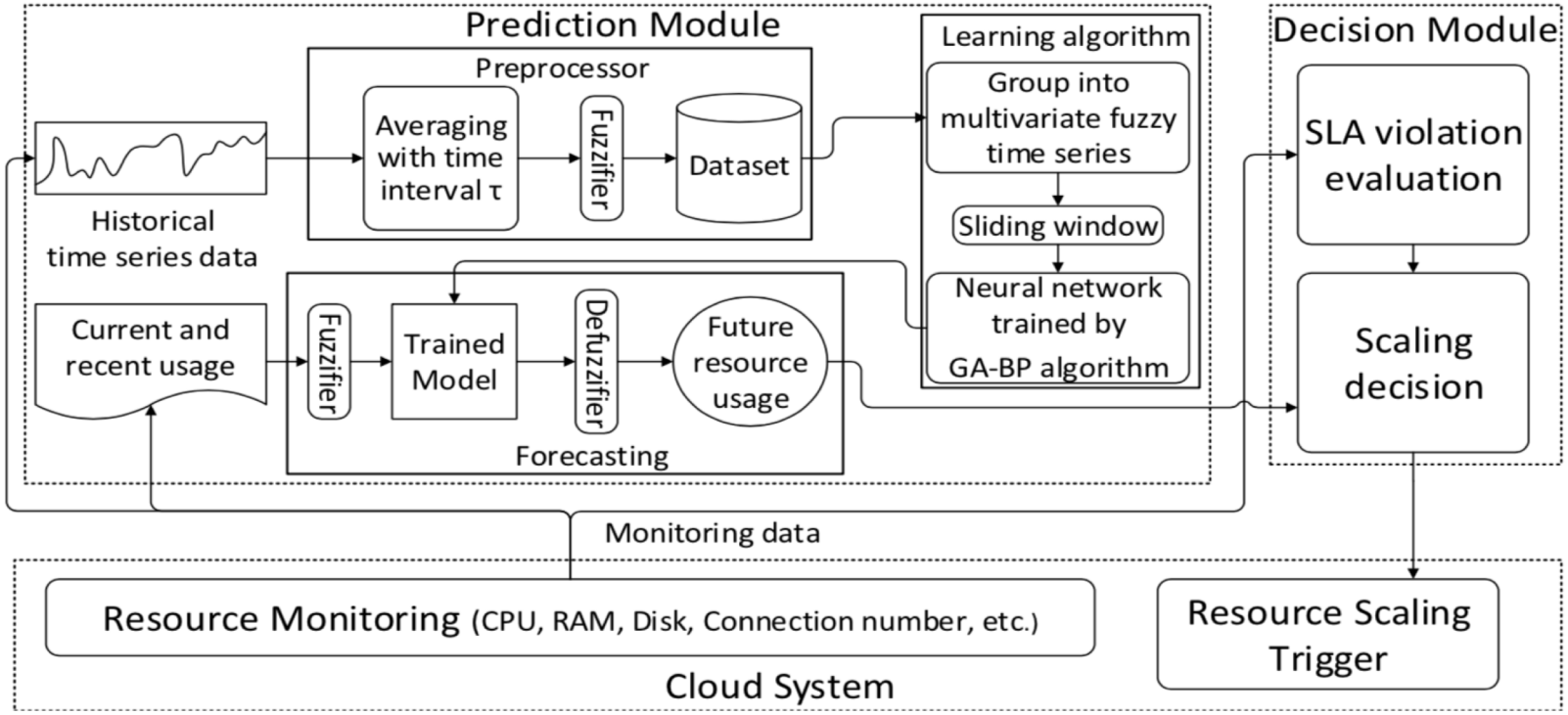


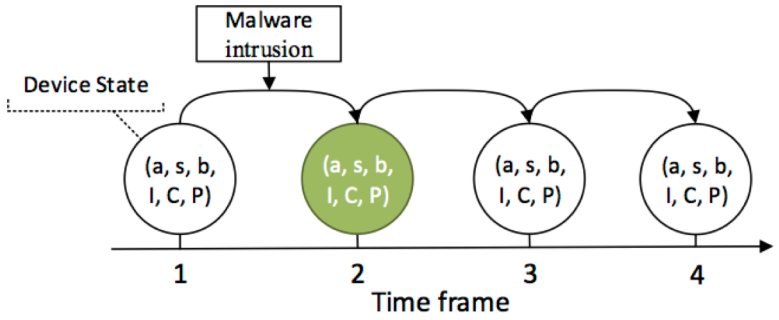
ML approaches

- Generative e.g. Naïve Bayes
- Discriminative e.g. Logistic Regression
- Neural Networks: FFNN, FCNN, DNN
- Deep Learning: RNN, LSTM, GRU

Example of data taxonomy

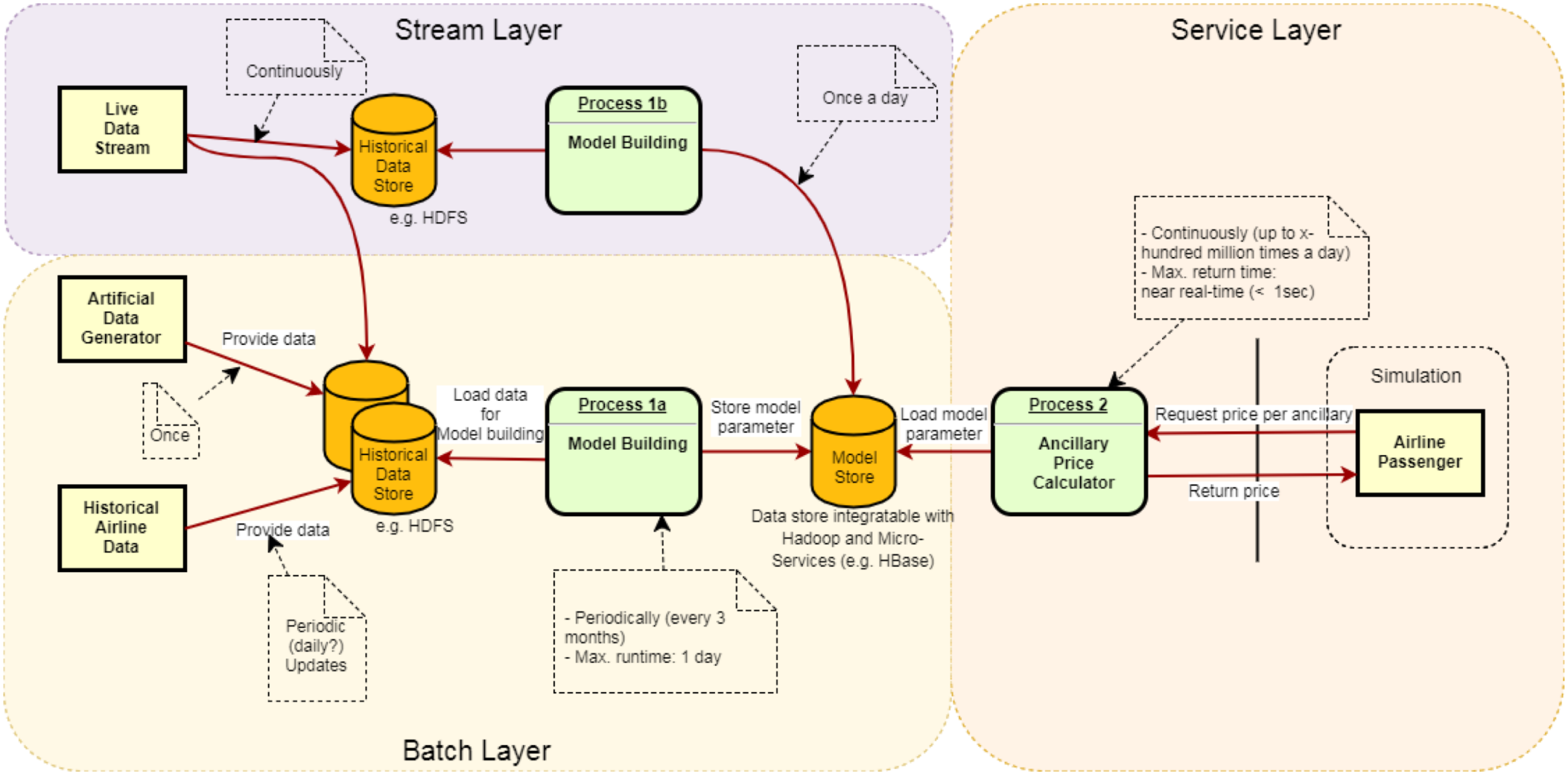




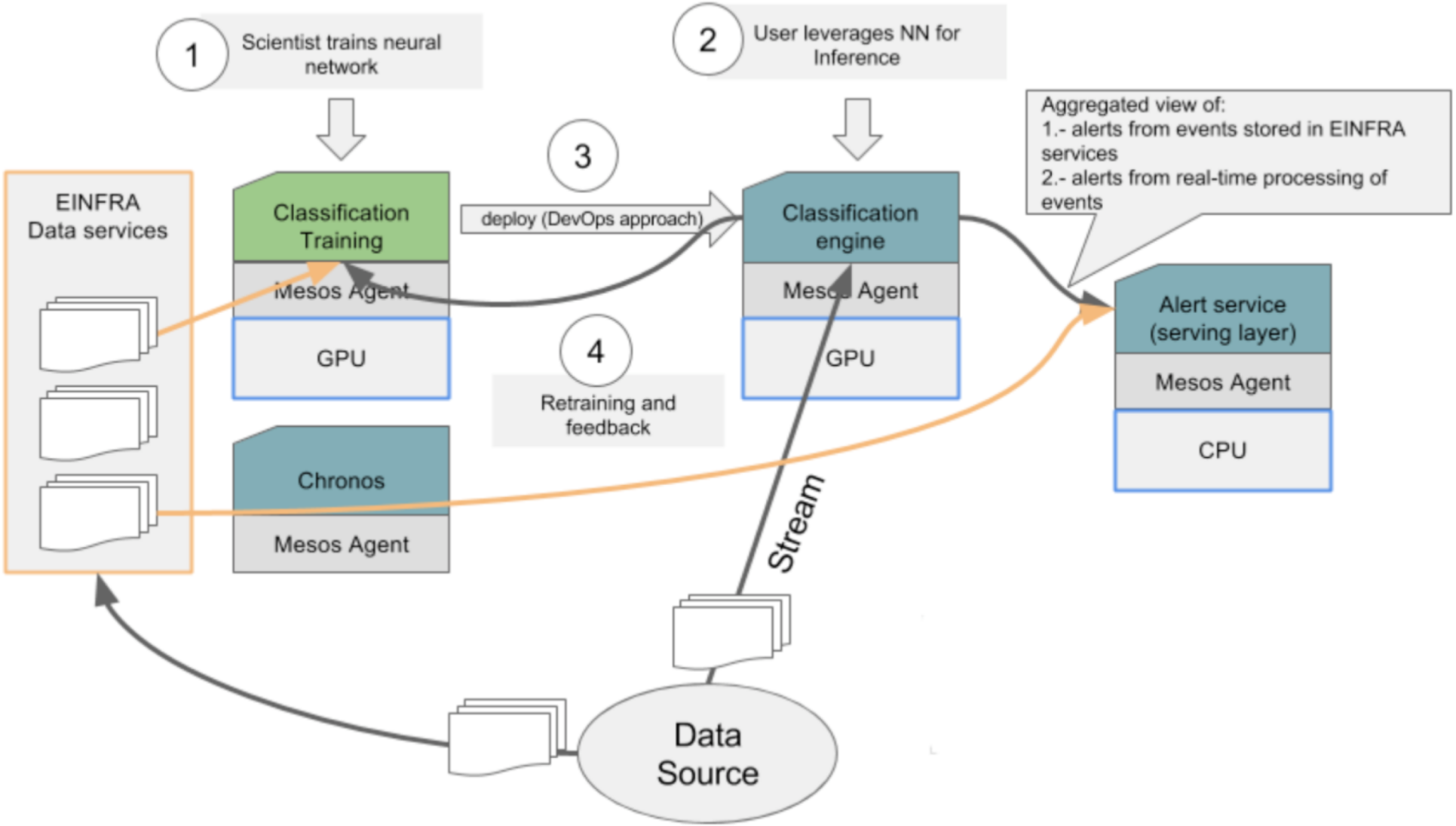


Example of data taxonomy

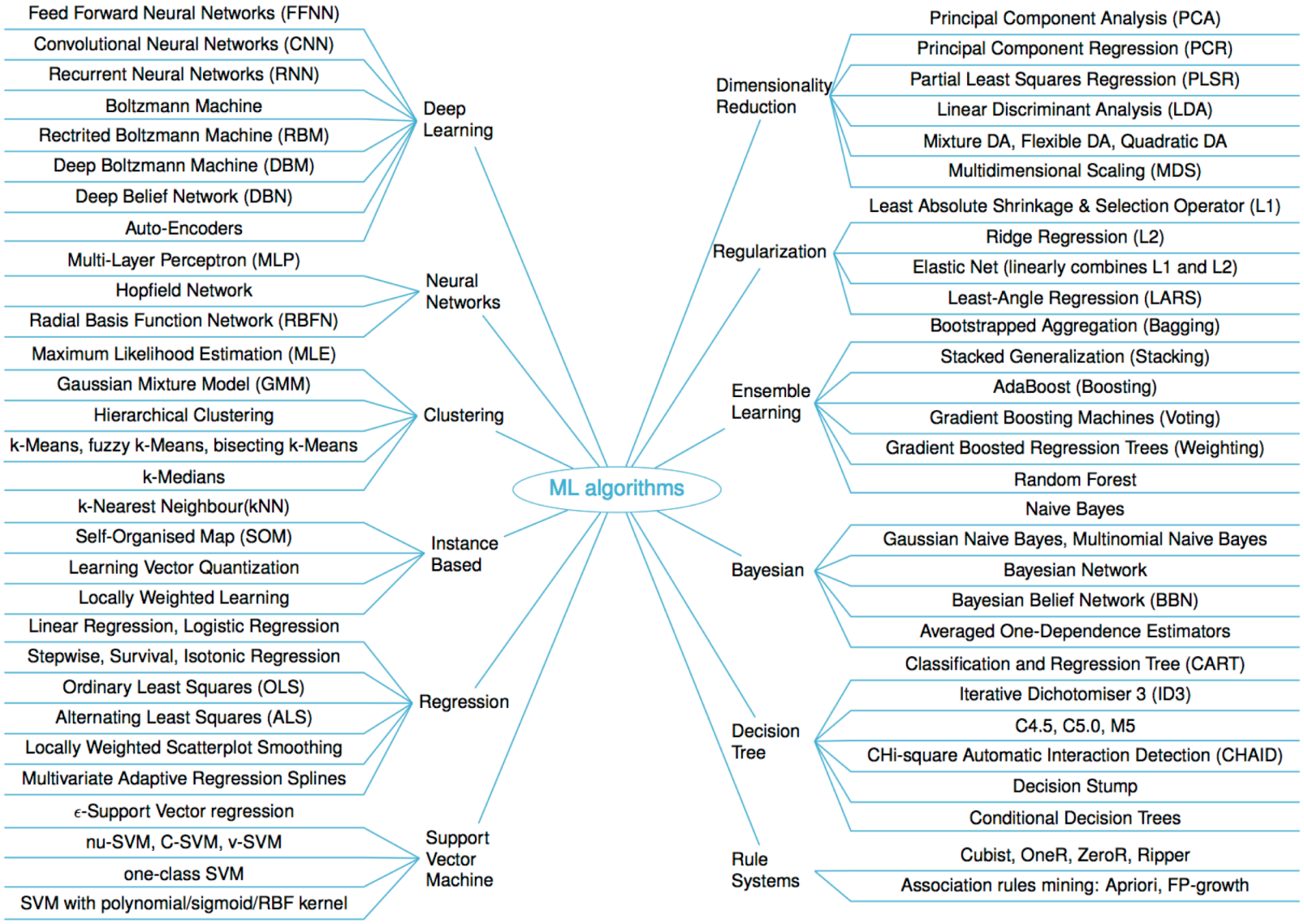
ML workflow – adaptive learning (Airbus)



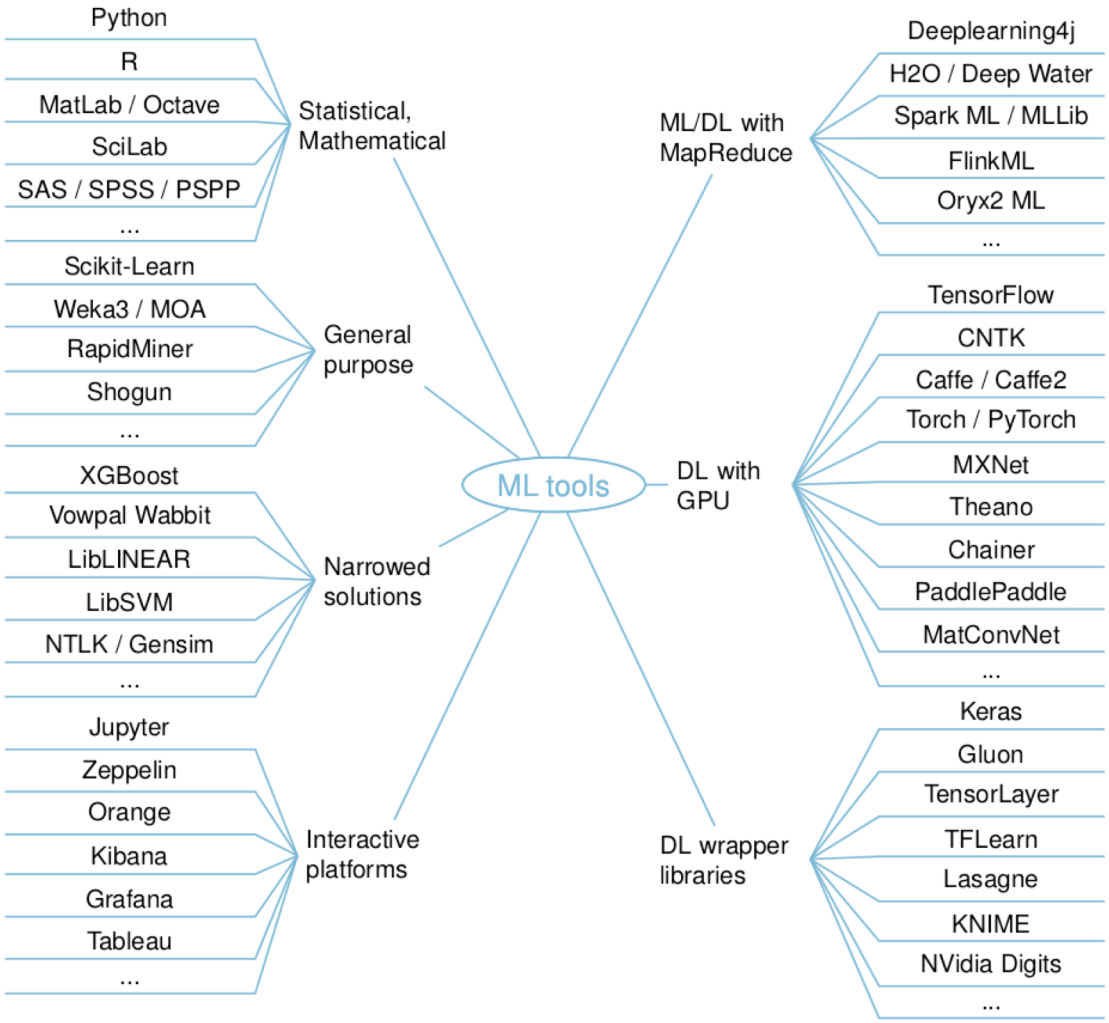
Deep UC #MODS - ML workflow

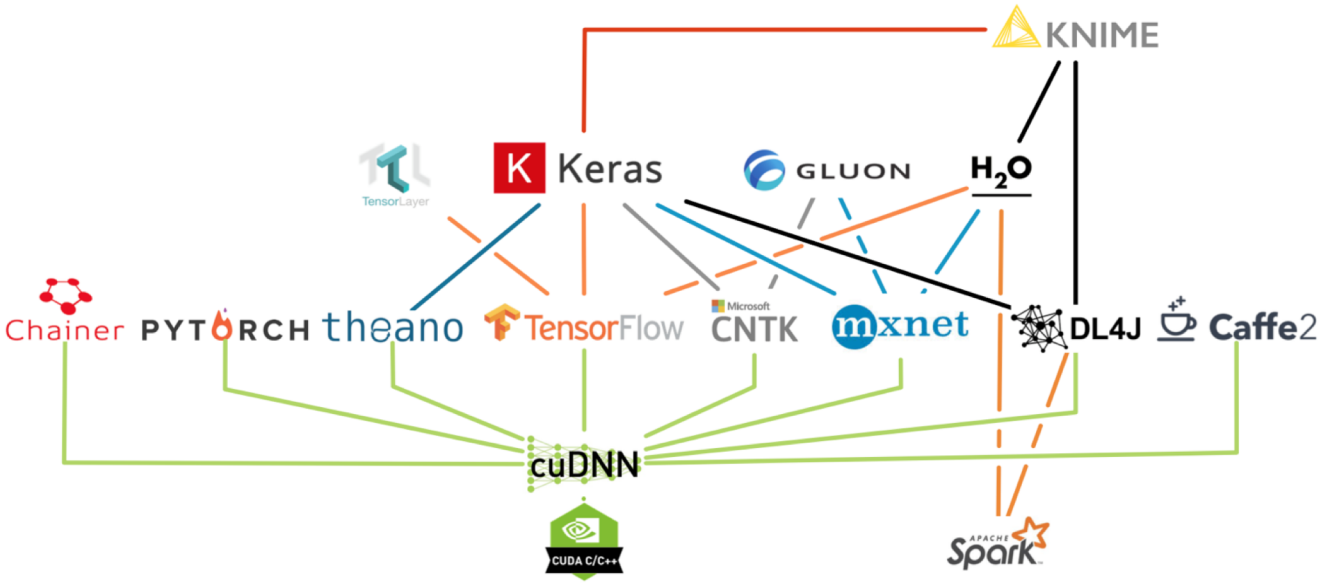


ML/DL methods - Overview

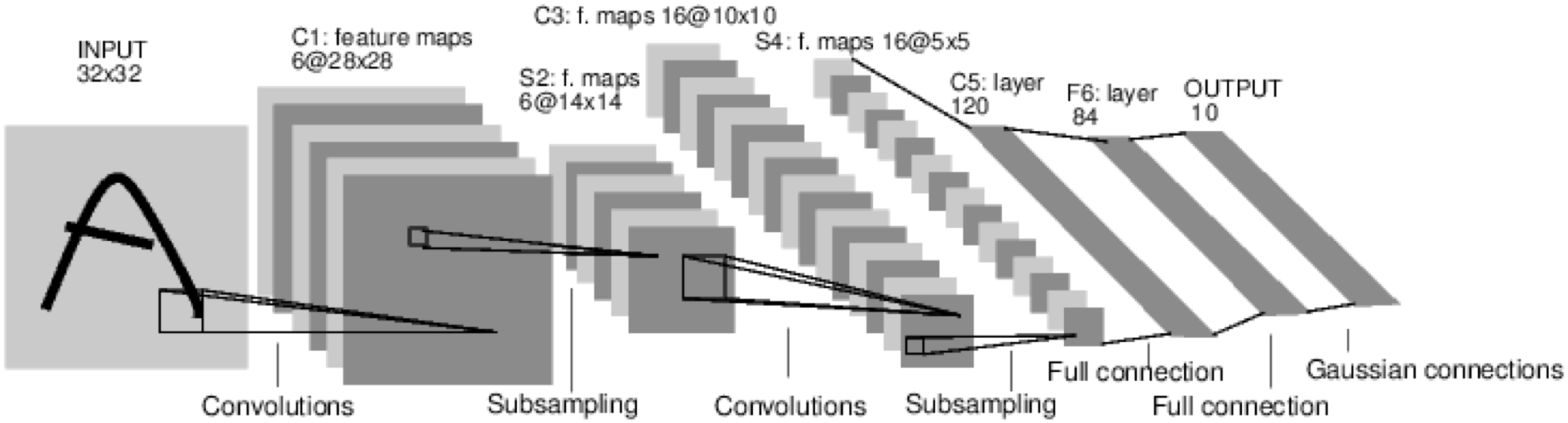


ML/DL tools - Overview





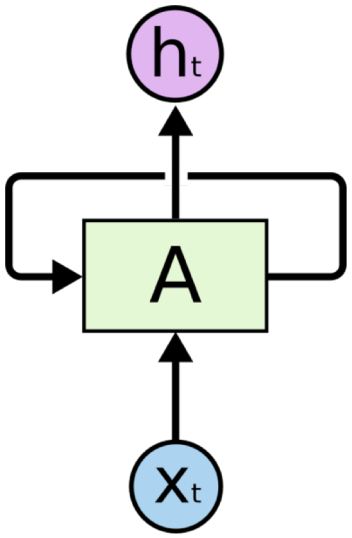
Deep Learning - CNN image processing

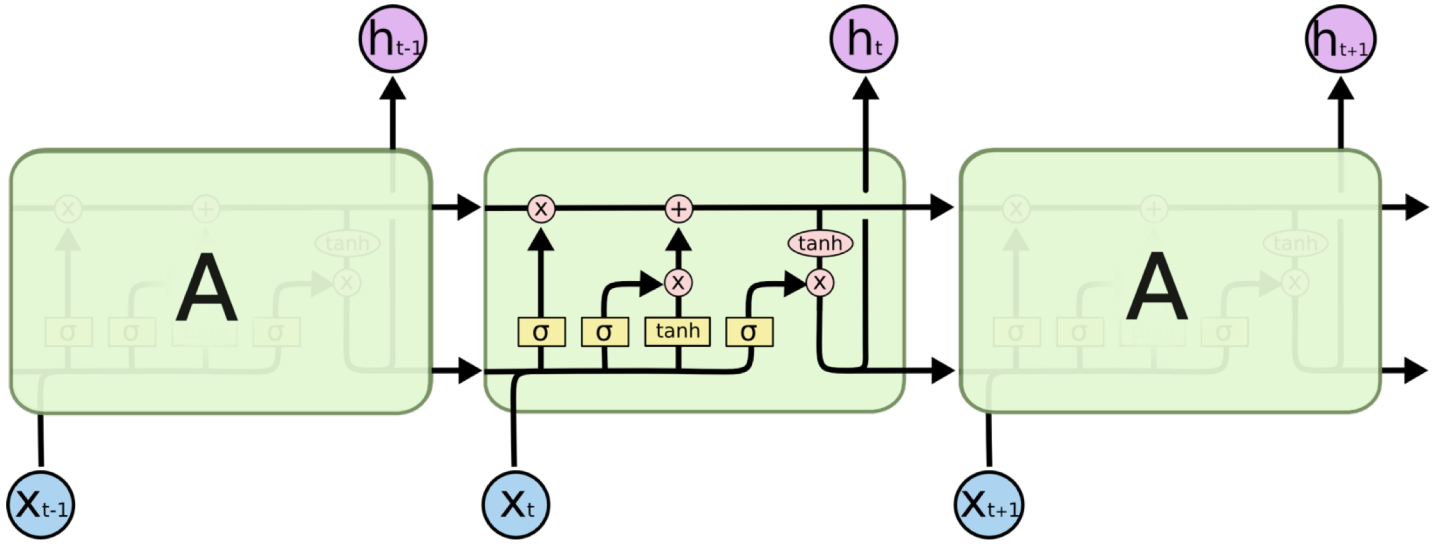


- Without feature engineering (oh, how good, really?)
  - Yes, images have the same size
  - No … ah …
- Data pre-processing and post-processing (still?)
  - Yes, image preprocessing from raw data e.g. maps or photos,
  - Image transformation … computer vision …

Deep Learning - RNN (LSTM/GRU block)

- Idea: information persistence, classification of events in sequences
- RNN has a loop in it, they might be able to connect previous information to the present task e.g. NLP, sensitive context, video frames
- A lot of news and unknown → a lot of self-study !!!

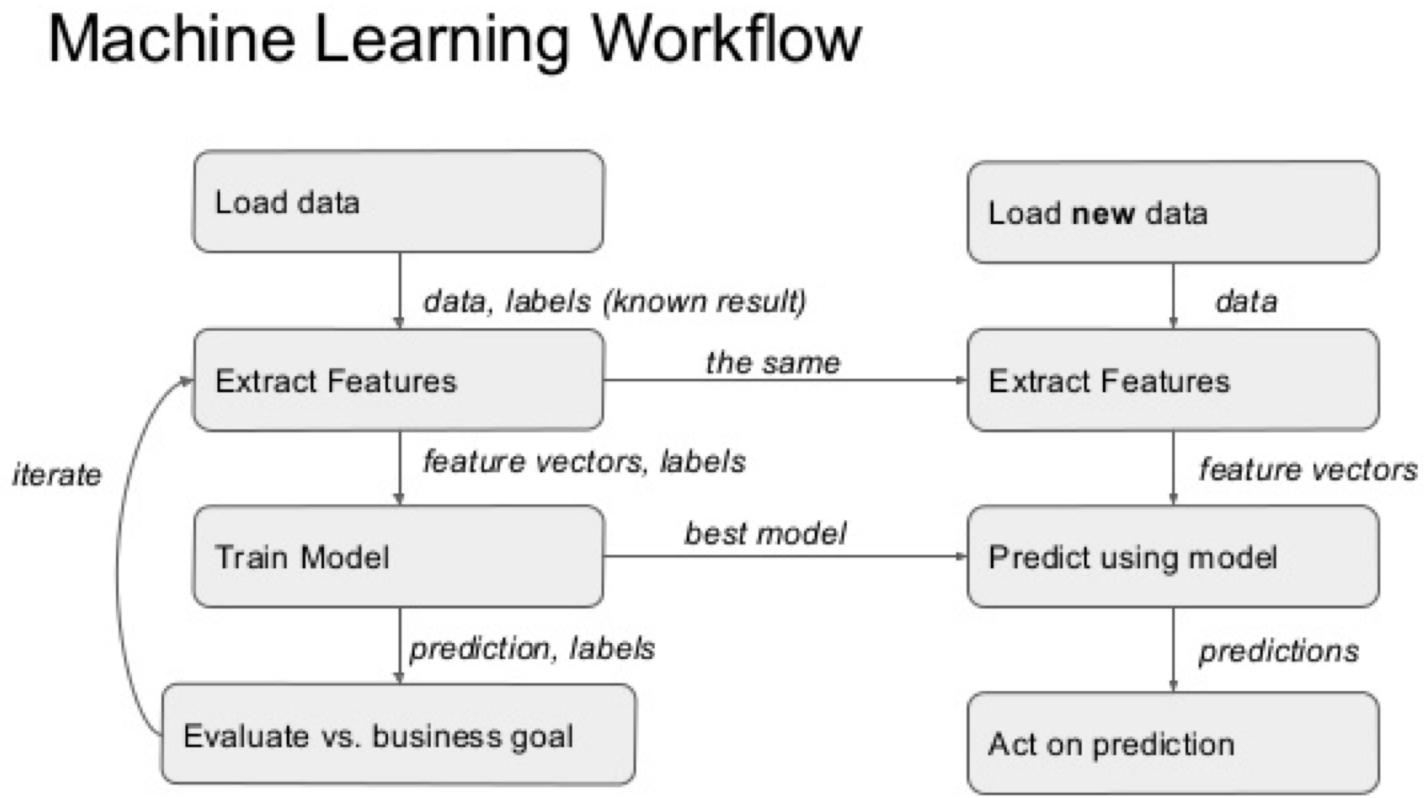




Deep #MODS → solutions

- **LSTM (vanilla, stacked, bidirectional), GRU for multivariate**
  - **DL swears better accuracy than NNs**
  - **Here is a news that CNN works fine also for that**
- Statistic and traditional ML certainly works
  - Unsupervised: outlier detection, clustering (k-mean, kNN), …
  - Supervised: proactive, reactive
  - Semi-supervised (active learning)
- Many solutions for many monitoring as well as cybersecurity issues
- **Data wrangle, a lot of testing are unavoidable**

Integration with underlying IDS (Bro)



Thank you
for your attention

Giang Nguyen

giang.ui@savba.sk