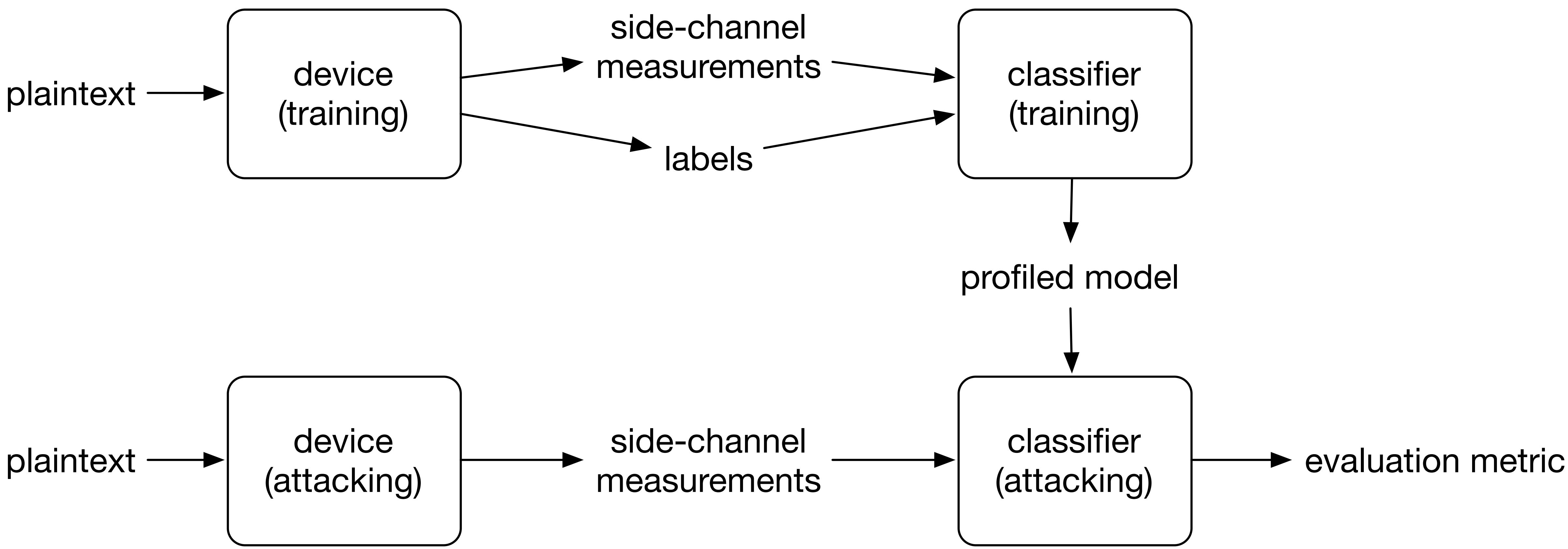
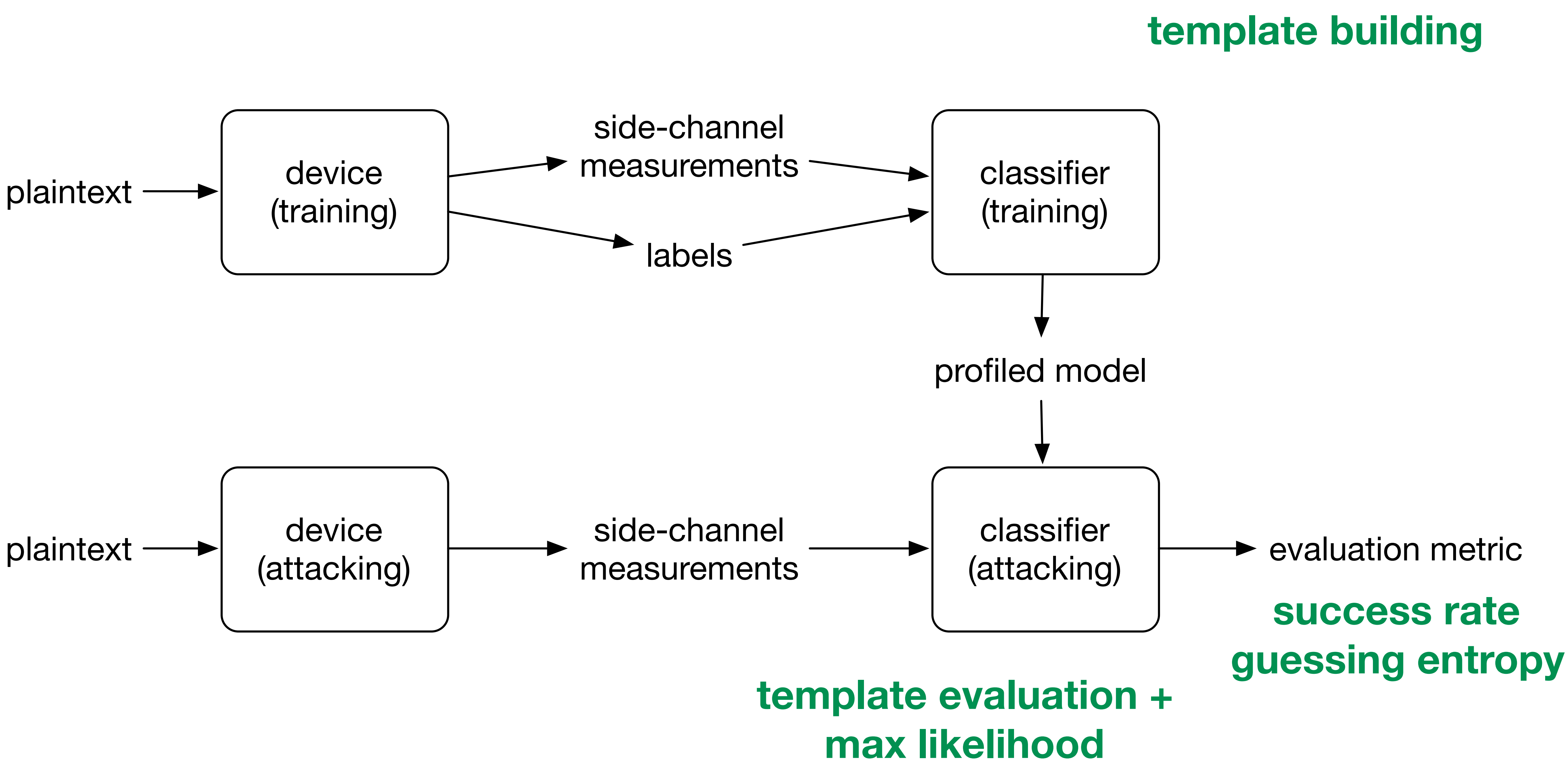
The Curse of Class Imbalance and Conflicting Metrics with Machine Learning for Side-channel Evaluations

Stjepan Picek, Annelie Heuser, Alan Jovic, Shivam Bhasin, and Francesco Regazzoni

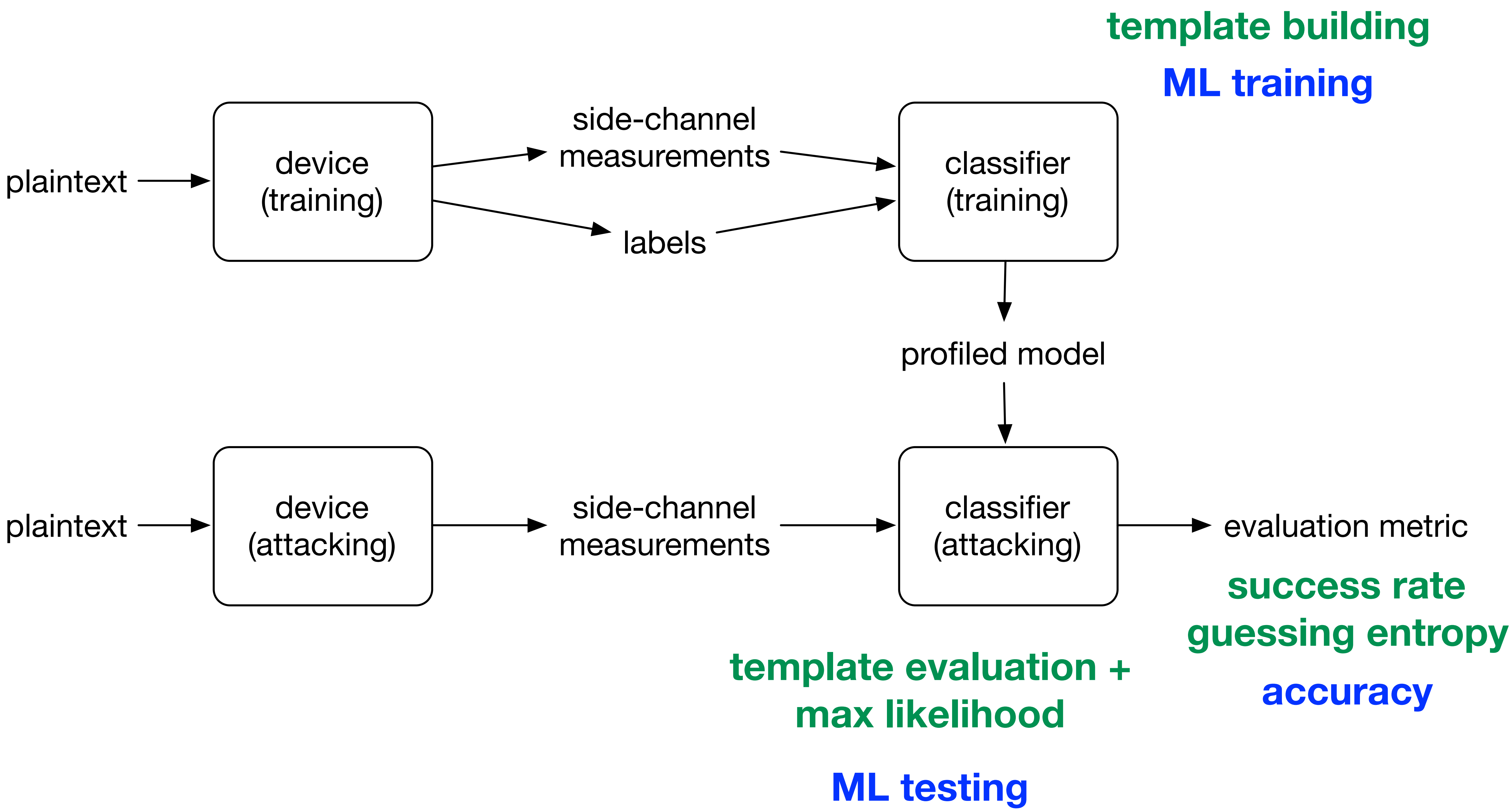
Big Picture


plaintext
device
(training)
side-channel
measurements
labels
classifier
(training)
profiled model
plaintext
device
(attacking)
side-channel
measurements
classifier
(attacking)
evaluation metric

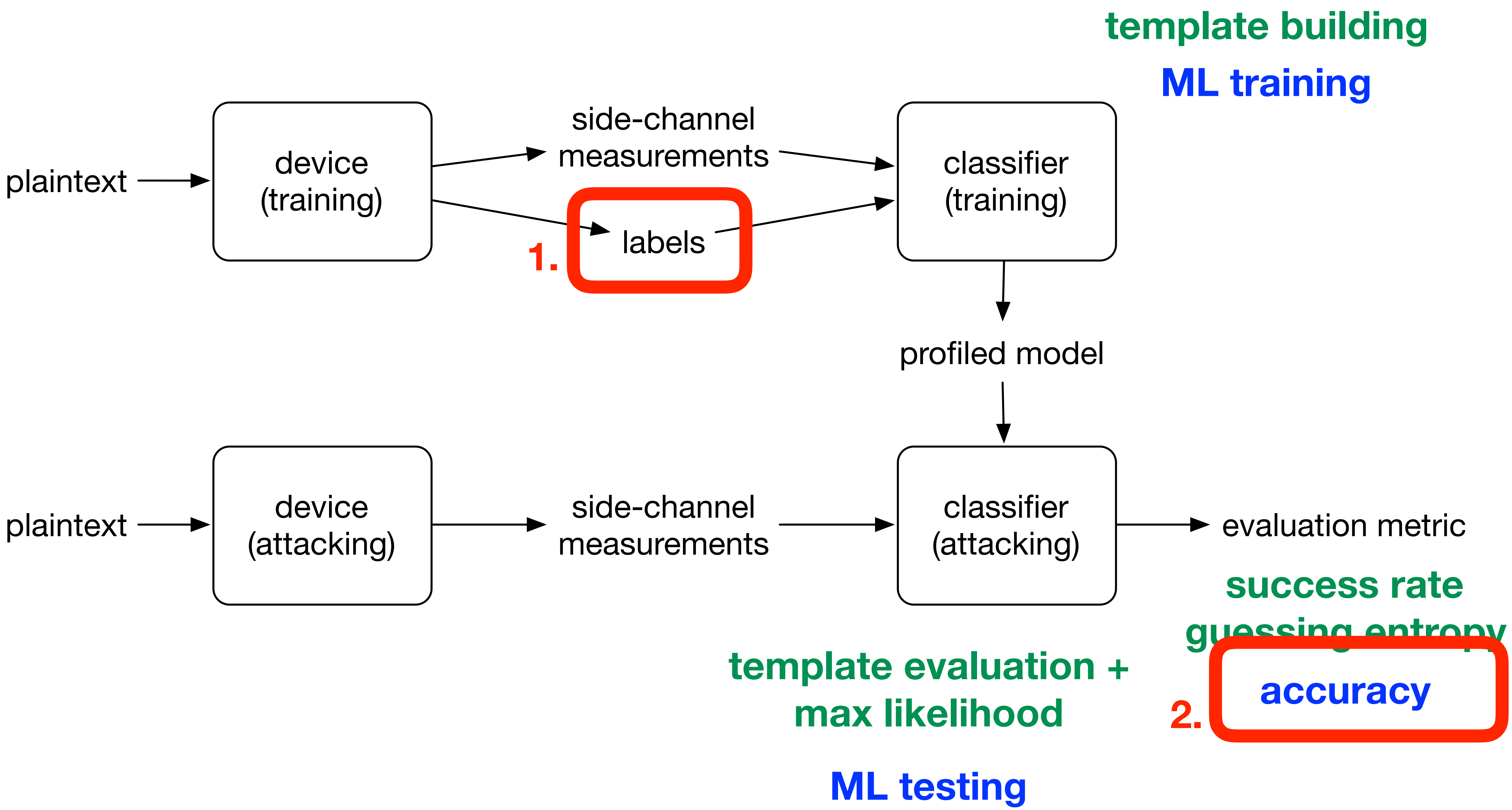
Big Picture


template building
plaintext
device
(training)
side-channel
measurements
labels
classifier
(training)
profiled model
plaintext
device
(attacking)
side-channel
measurements
classifier
(attacking)
evaluation metric
success rate
guessing entropy
template evaluation +
max likelihood

Big Picture


template building
ML training
plaintext
device (training)
side-channel measurements
labels
classifier (training)
profiled model
plaintext
device (attacking)
side-channel measurements
classifier (attacking)
evaluation metric
success rate
guessing entropy
accuracy
template evaluation + max likelihood
ML testing

Big Picture


template building
ML training
plaintext
device
(training)
side-channel
measurements
classifier
(training)
1.
labels
profiled model
plaintext
device
(attacking)
side-channel
measurements
classifier
(attacking)
evaluation metric
success rate
guessing entropy
template evaluation +
max likelihood
2.
accuracy
ML testing

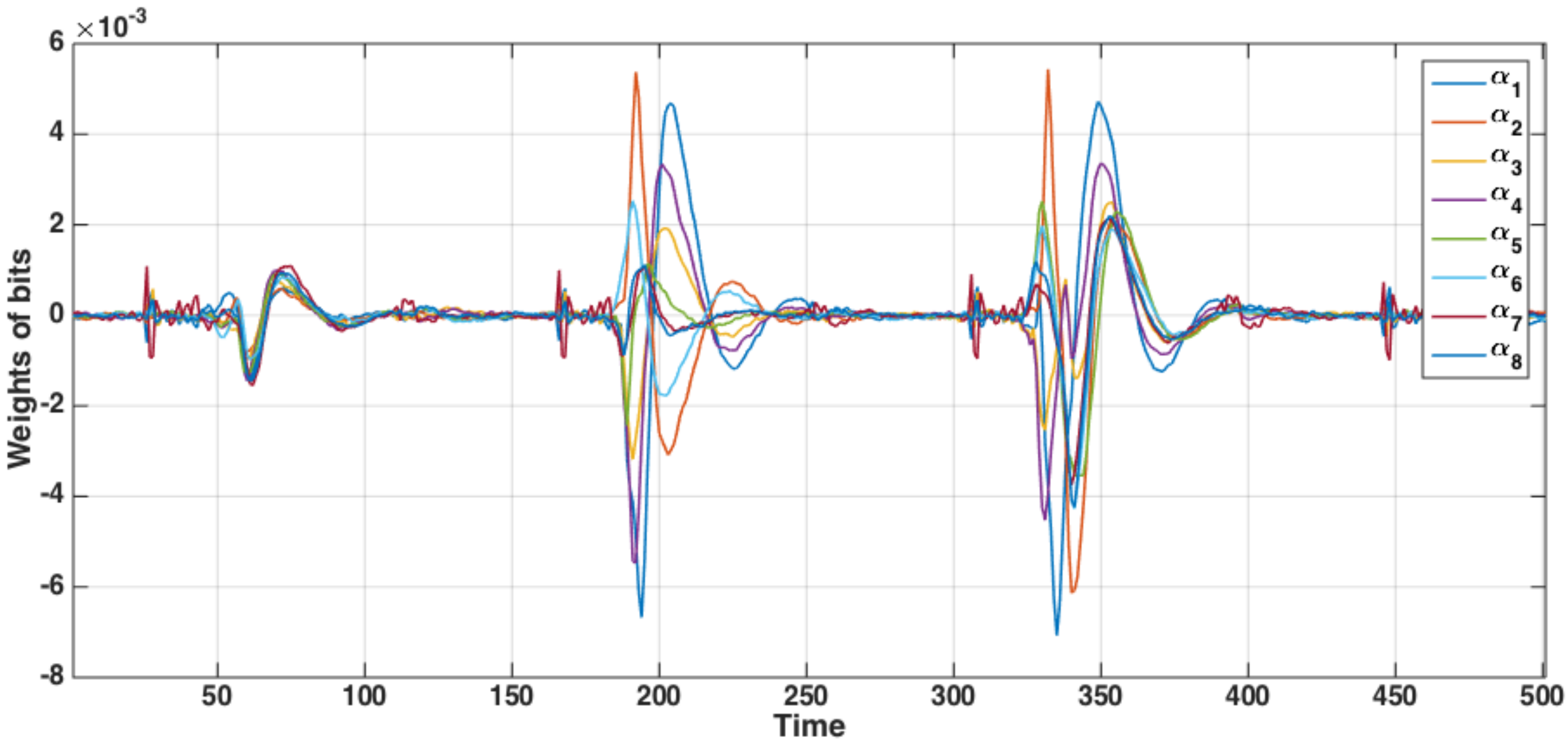
Labels

- typically: intermediate states computed from plaintext and keys
- Hamming weight (distance) leakage model commonly used
- problem: introduces imbalanced data
- for example, occurrences of Hamming weights for all possible 8-bit values:

HW value	0	1	2	3	4	5	6	7	8
Occurrences	1	8	28	56	70	56	28	8	1

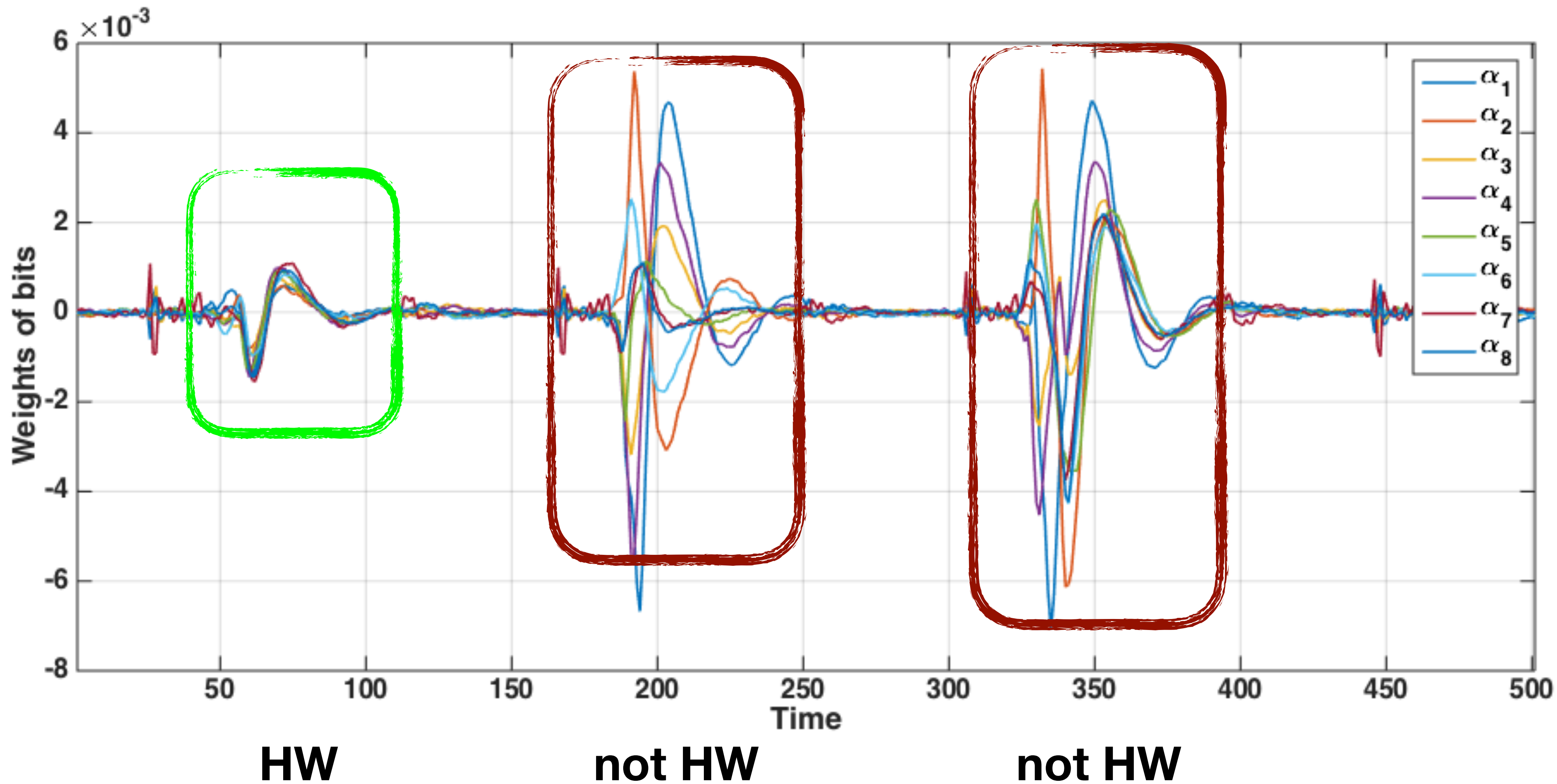
Why do we use HW?

- often does not reflect realistic leakage model



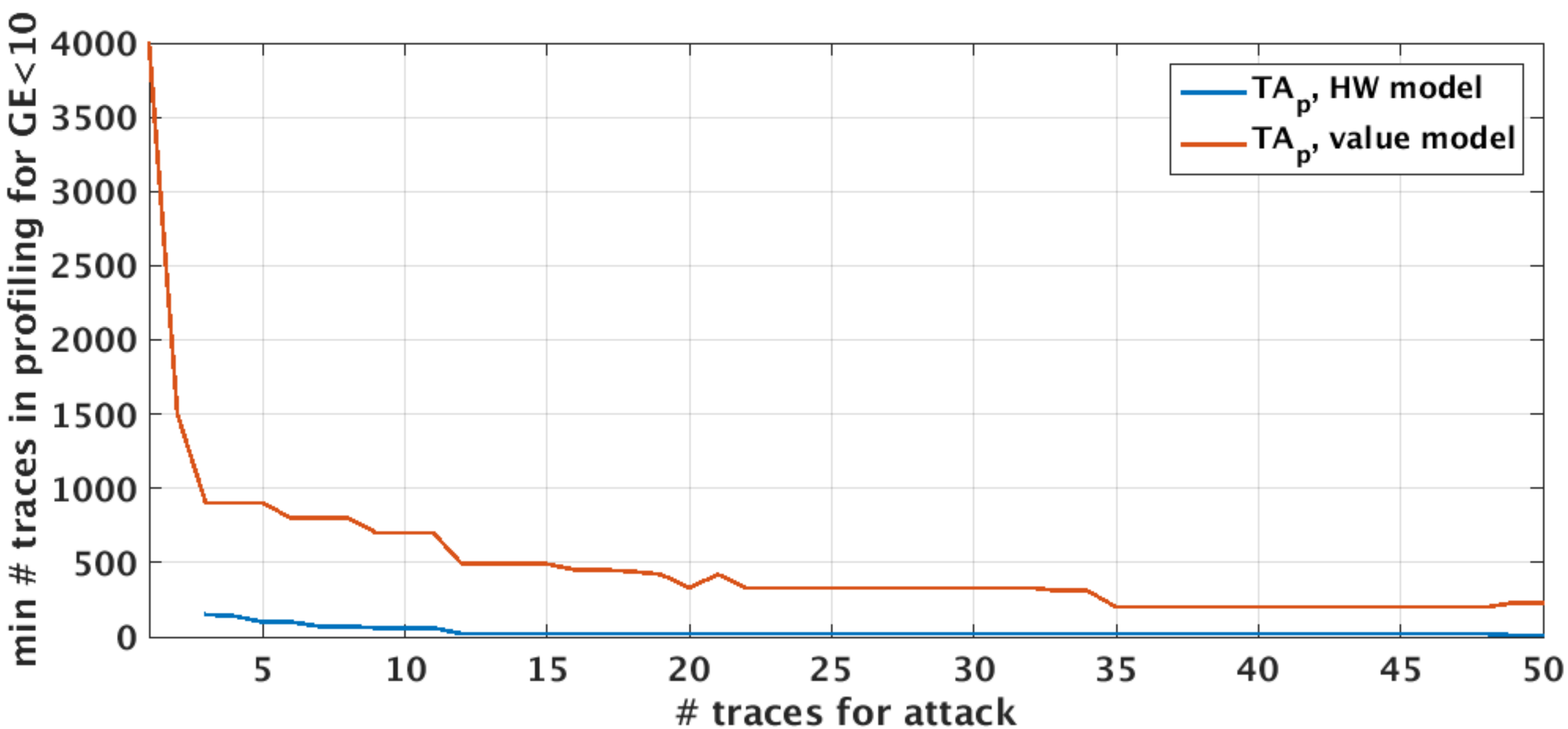
Why do we use HW?

- often does not reflect realistic leakage model



Why do we use HW?

- reduces the complexity of learning
- works (sufficiently good) in many scenarios for attacking



Why do we care about imbalanced data?

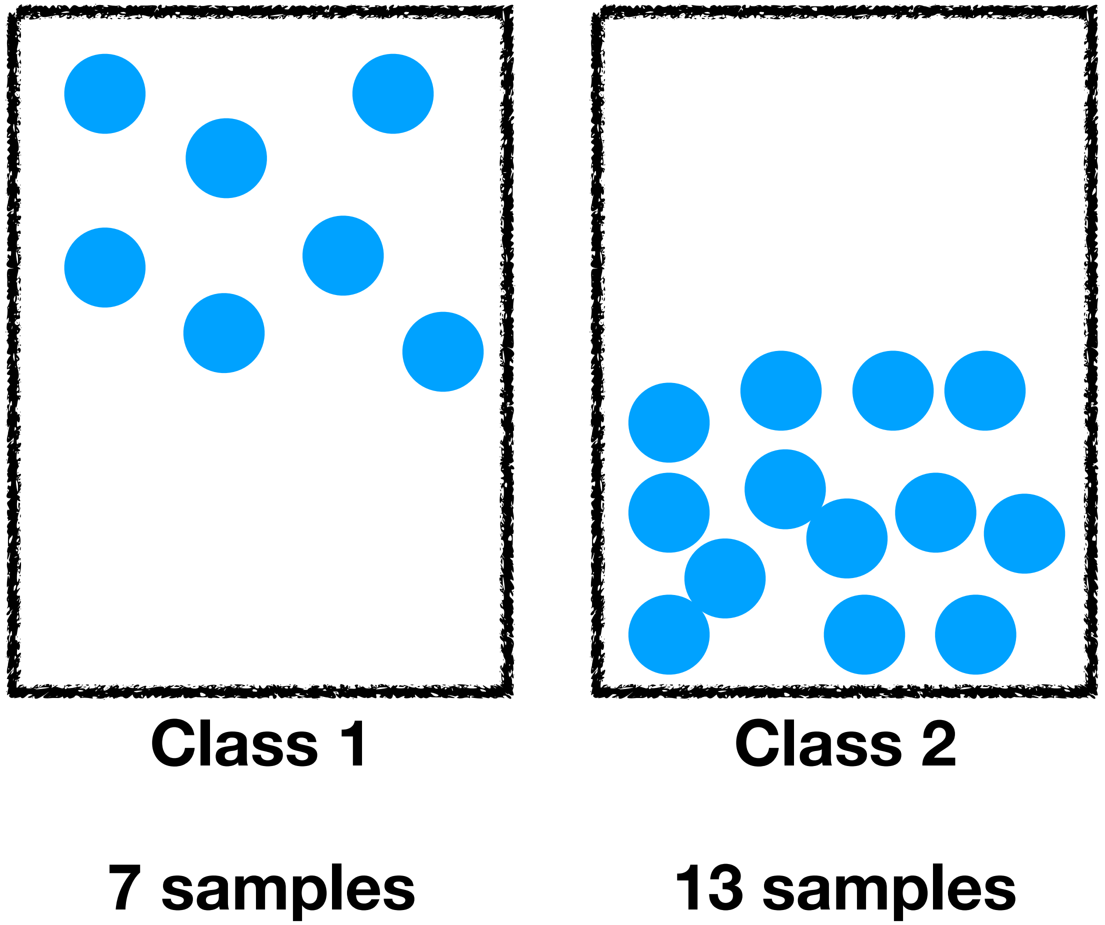
- most machine learning techniques rely on loss functions that are “designed” to maximise accuracy
- in case of high noise: predicting only HW class 4 gives accuracy of 27%
- but is not related to secret key value and therefore does not give any information for SCA

What to do?

- in this paper: transform dataset to achieve balancedness?
- how?
  - throw away data
  - add data
  - (or choose data before ciphering)

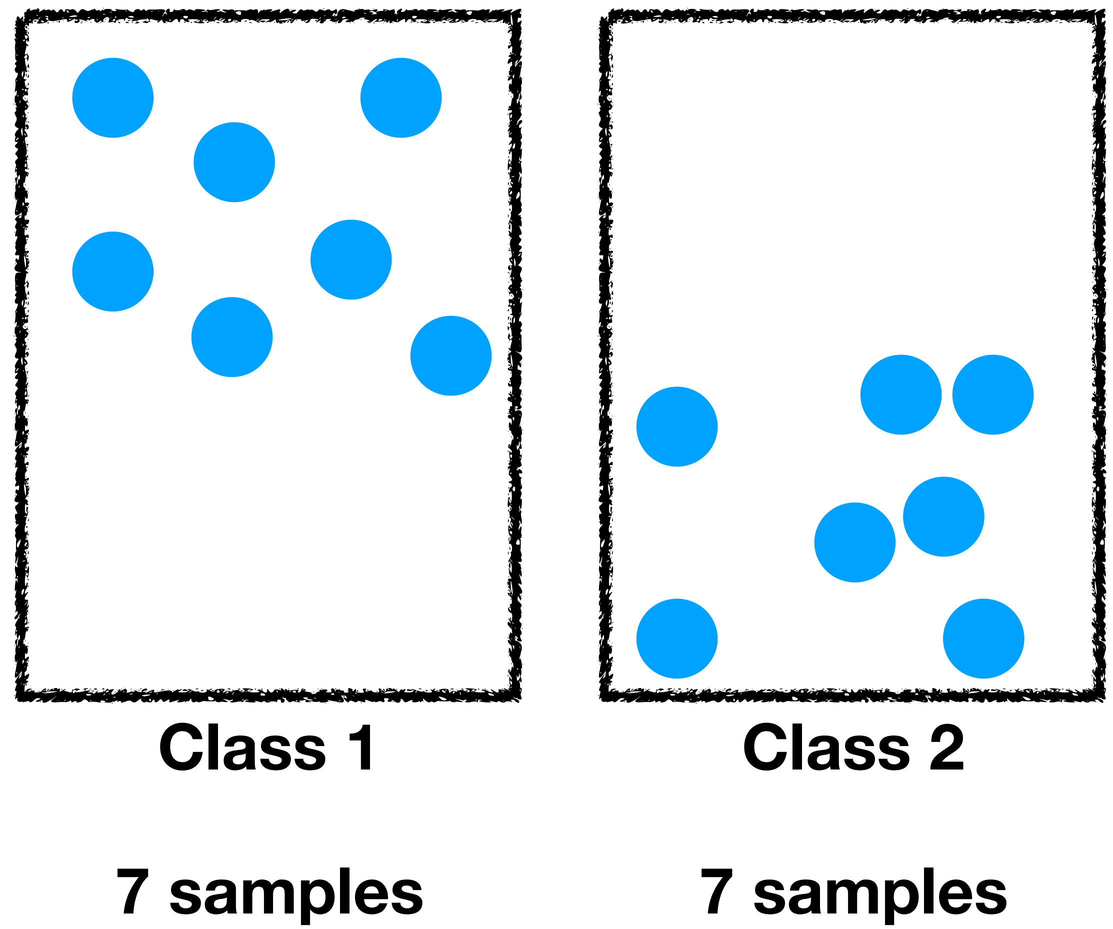
Random under sampling

- only keep # of samples equal to the least populated class
- binomial distribution: many unused samples



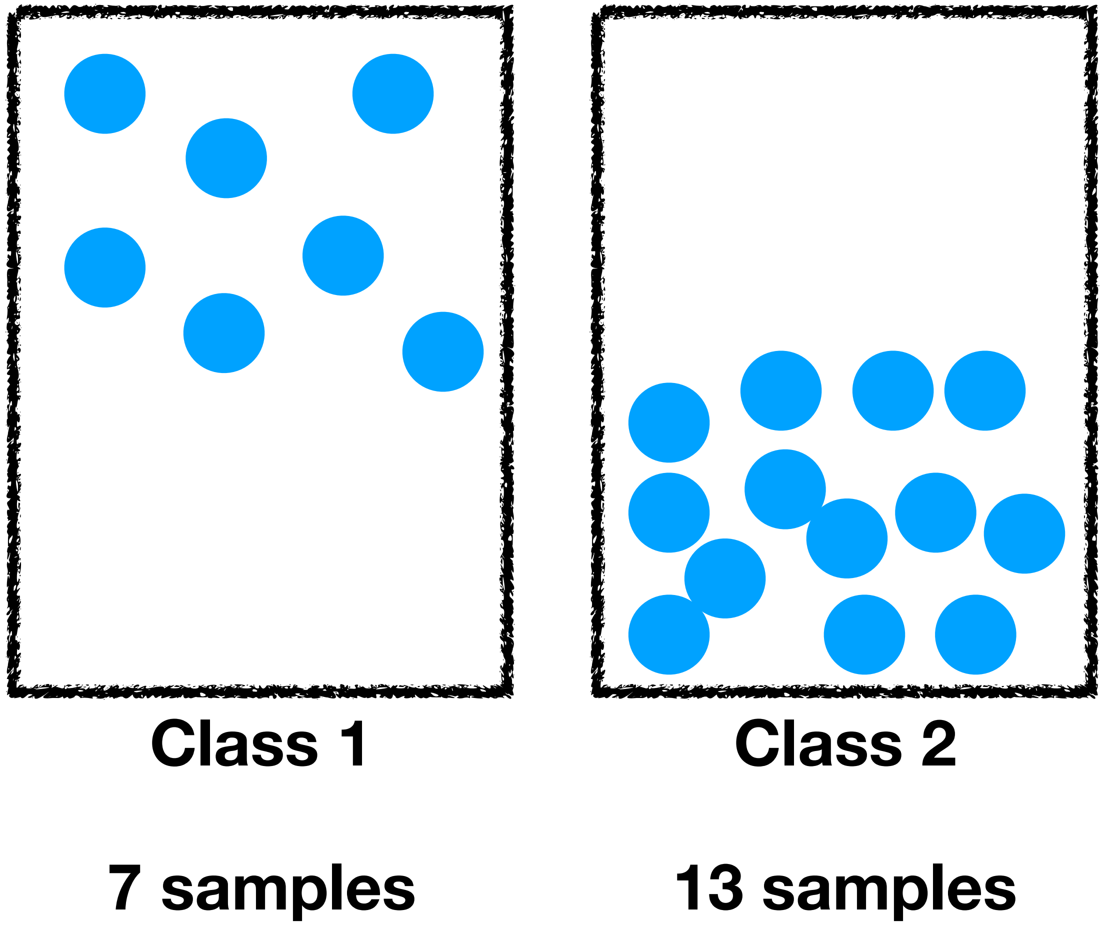
Random under sampling

- only keep # of samples equal to the least populated class
- binomial distribution: many unused samples



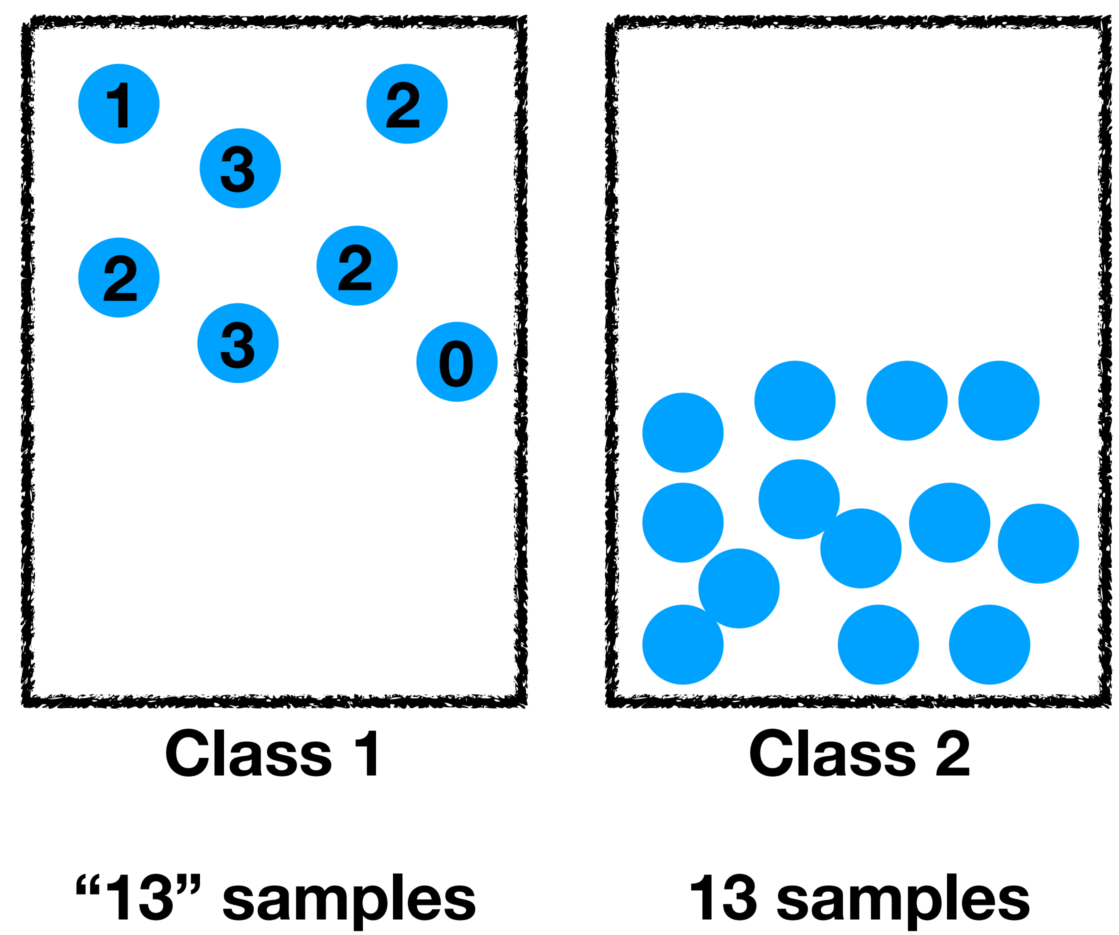
Random oversampling with replacement

- randomly selecting samples from the original dataset until amount is equal to largest populated
- simple method, in other context comparable to other methods
- may happen that some samples are not selected at all



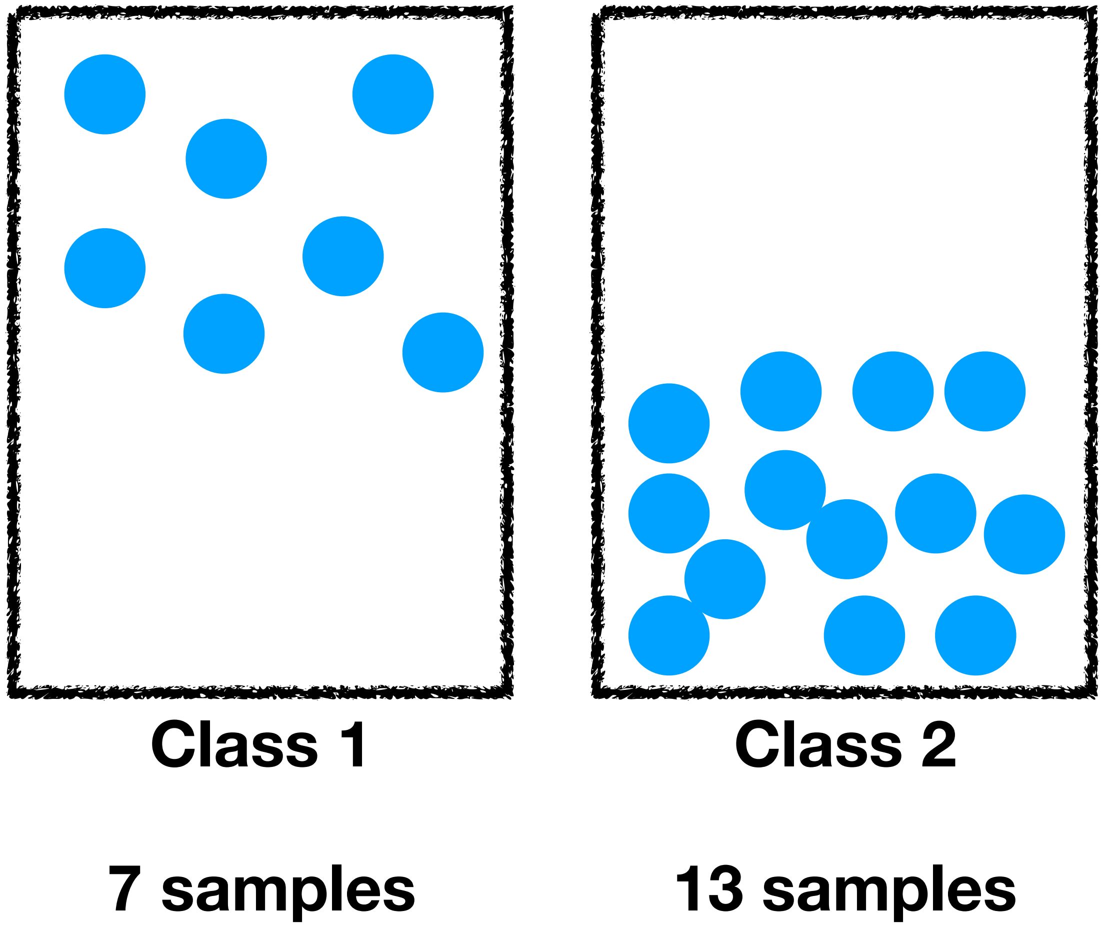
Random oversampling with replacement

- randomly selecting samples from the original dataset until amount is equal to largest populated
- simple method, in other context comparable to other methods
- may happen that some samples are not selected at all



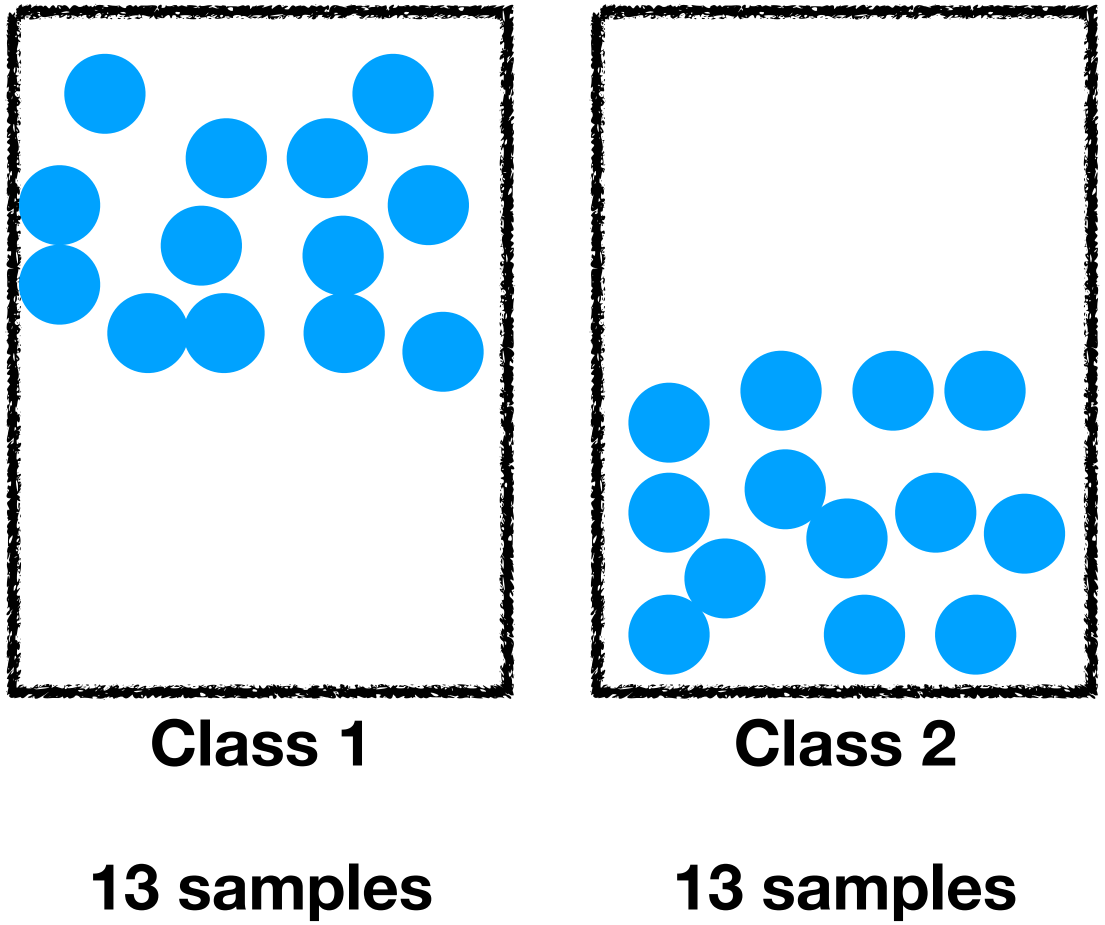
SMOTE

- synthetic minority oversampling technique
- generating synthetic minority class instances
- nearest neighbours are added (corresponding to Euclidean distance)



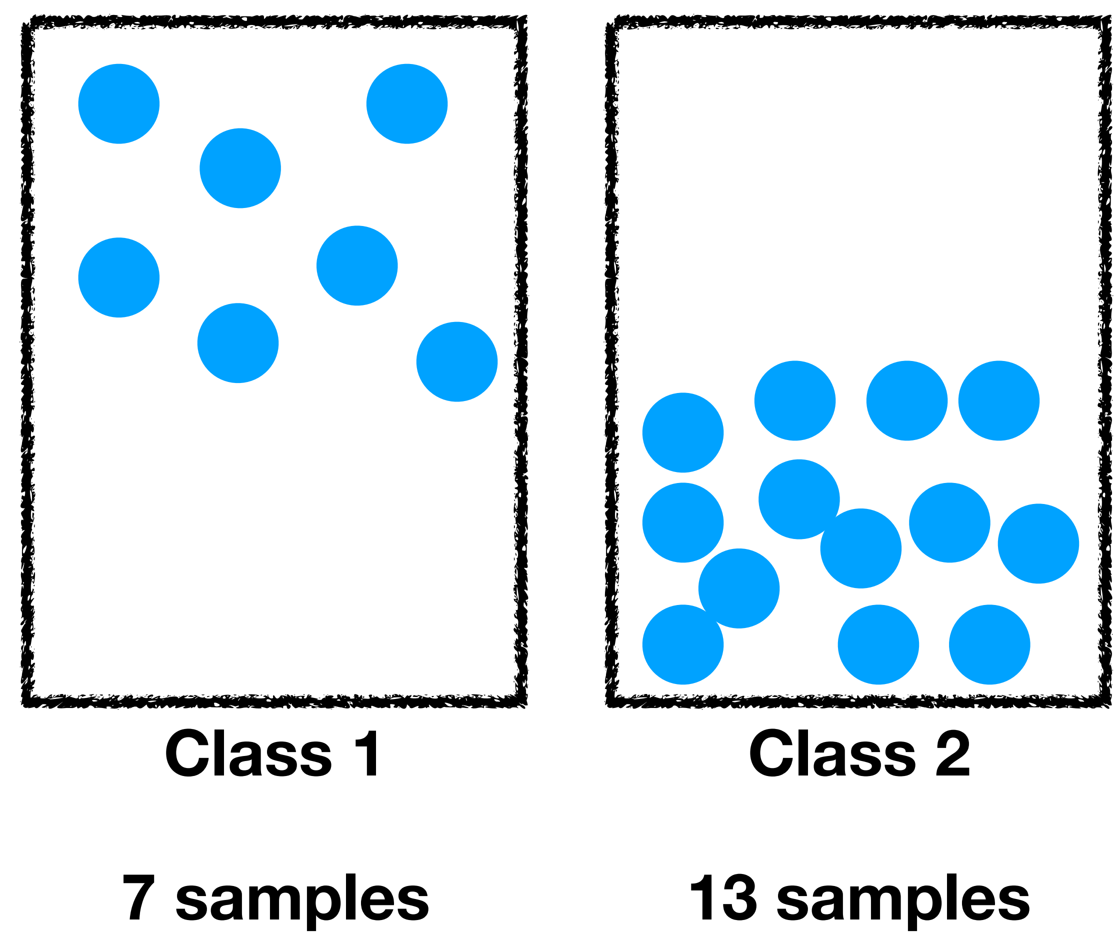
SMOTE

- synthetic minority oversampling technique
- generating synthetic minority class instances
- nearest neighbours are added (corresponding to Euclidean distance)



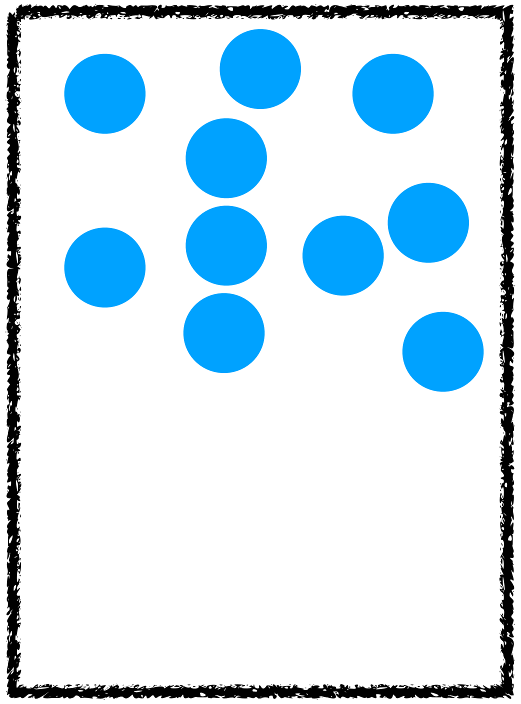
SMOTE+ENN

- Synthetic Minority Oversampling Technique with Edited Nearest Neighbor
- SMOTE + data cleaning
- oversampling + undersampling
- removes data samples whose class different from multiple neighbors



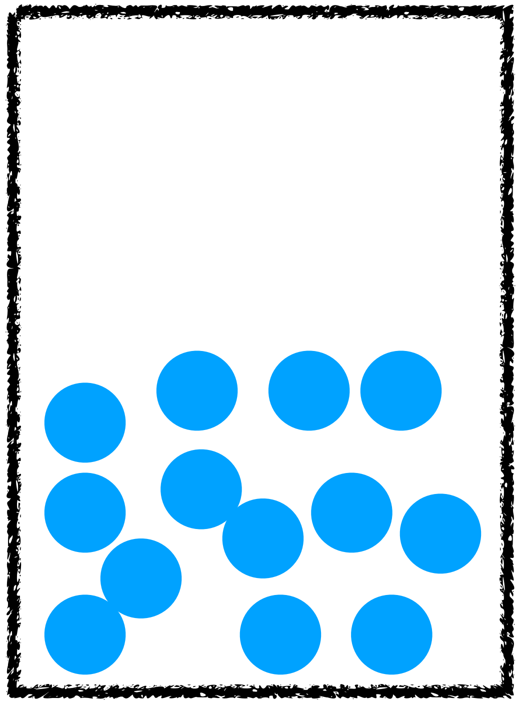
SMOTE+ENN

- Synthetic Minority Oversampling Technique with Edited Nearest Neighbor
- SMOTE + data cleaning
- oversampling + undersampling
- removes data samples whose class different from multiple neighbors



Class 1

10 samples



Class 2

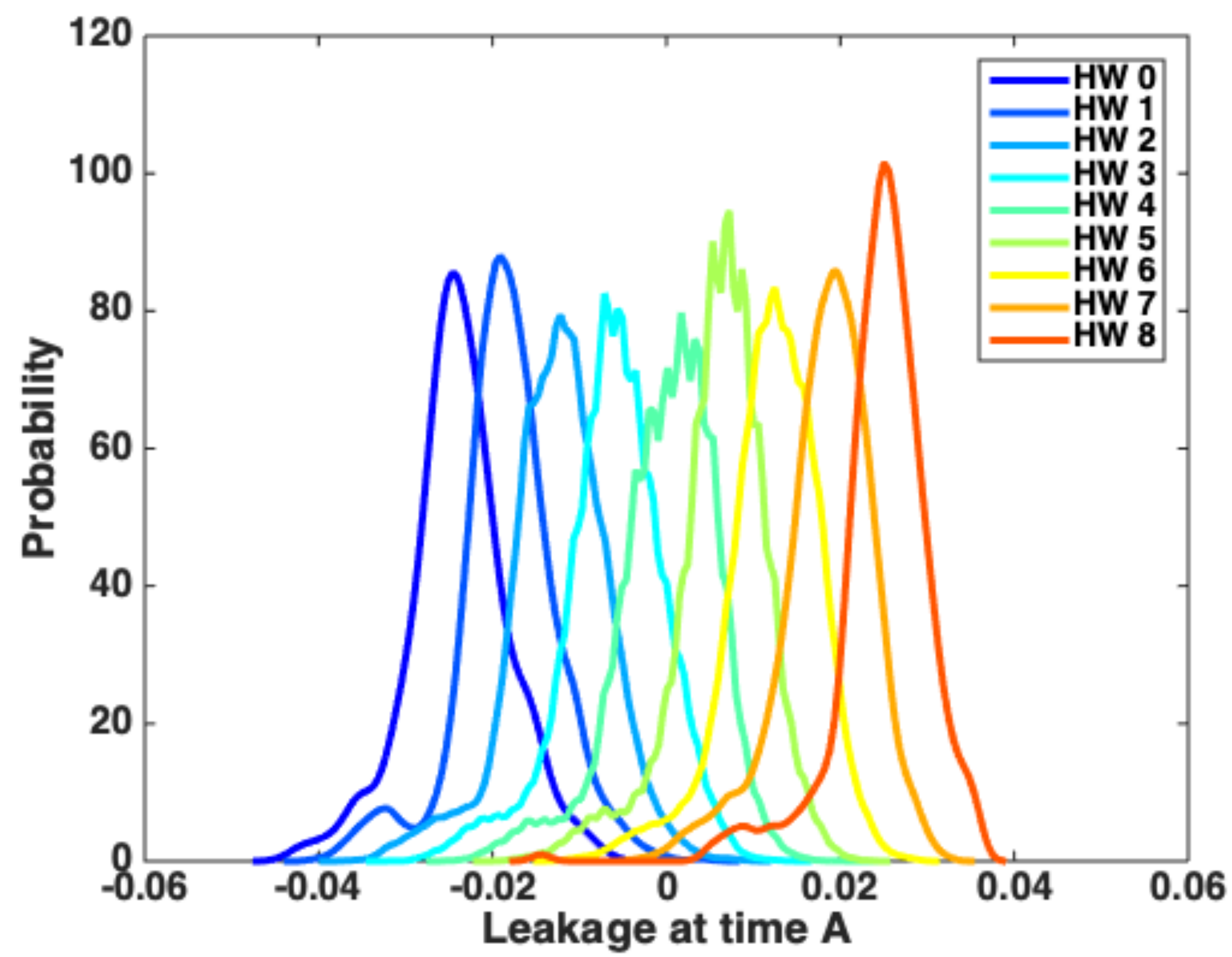
10 samples

Experiments

- in most experiments SMOTE most effective
- data argumentation without any specific knowledge about the implementation / dataset / distribution to balance datasets
- varying number of training samples in the profiling phase
  - Imbalanced: 1k, 10k, 50k
  - SMOTE: (approx) 5k, 24k, 120k

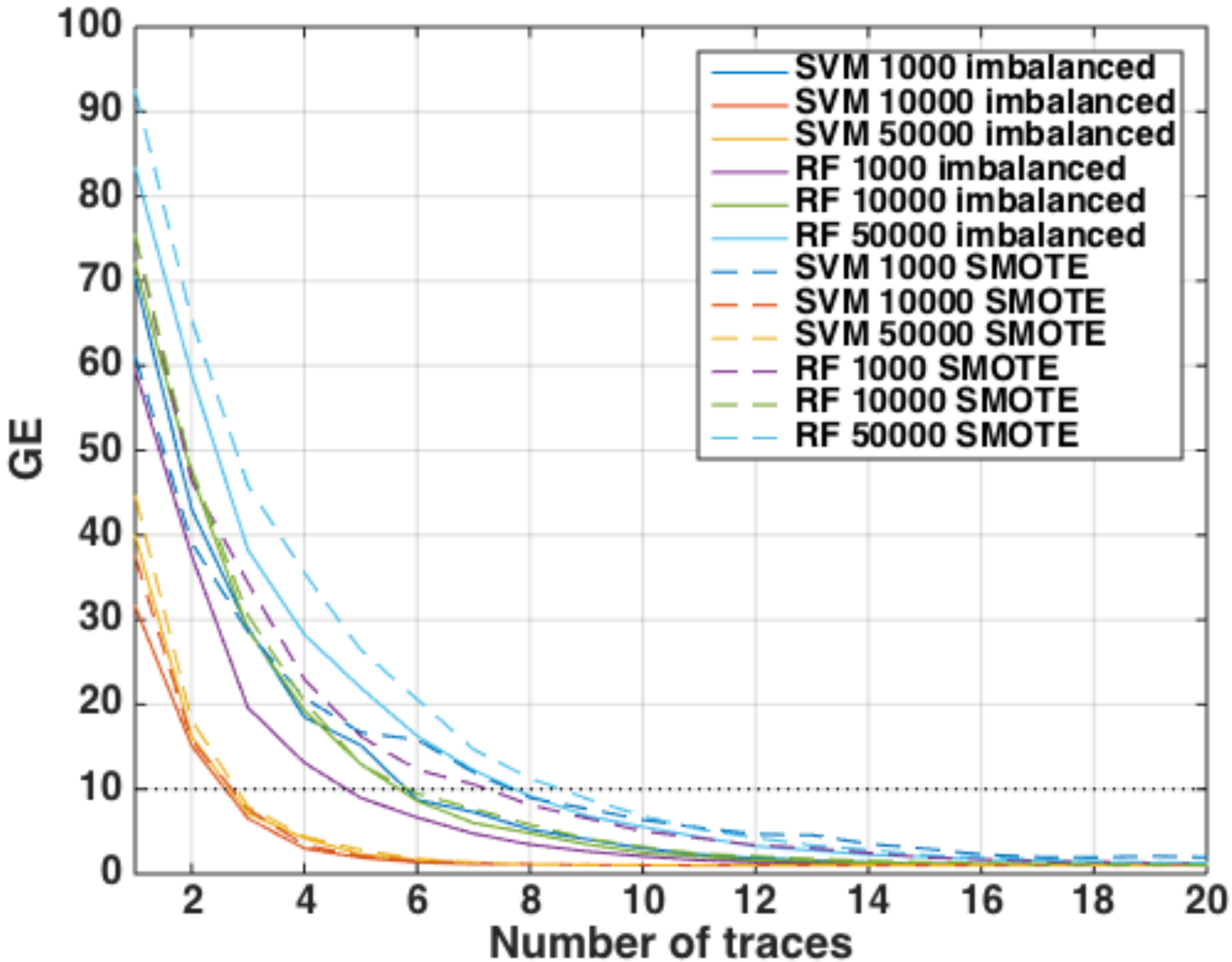
Dataset 1

- low noise dataset - DPA contest v4 (publicly available)
- Atmel ATMega-163 smart card connected to a SASEBO-W board
- AES-256 RSM
  (Rotating SBox Masking)
- in this talk:
  mask assumed known



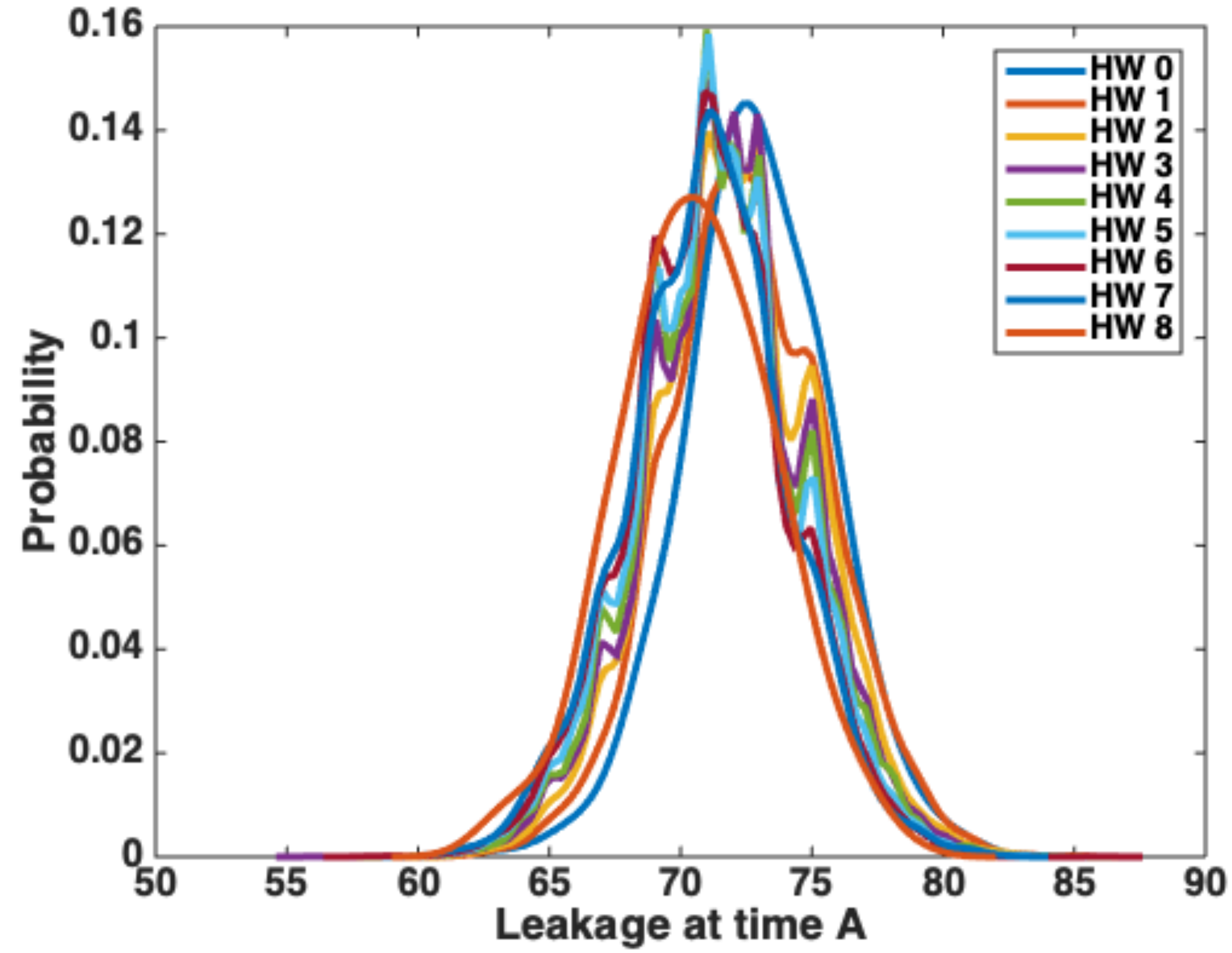
Data sampling techniques

- dataset 1: low noise unprotected



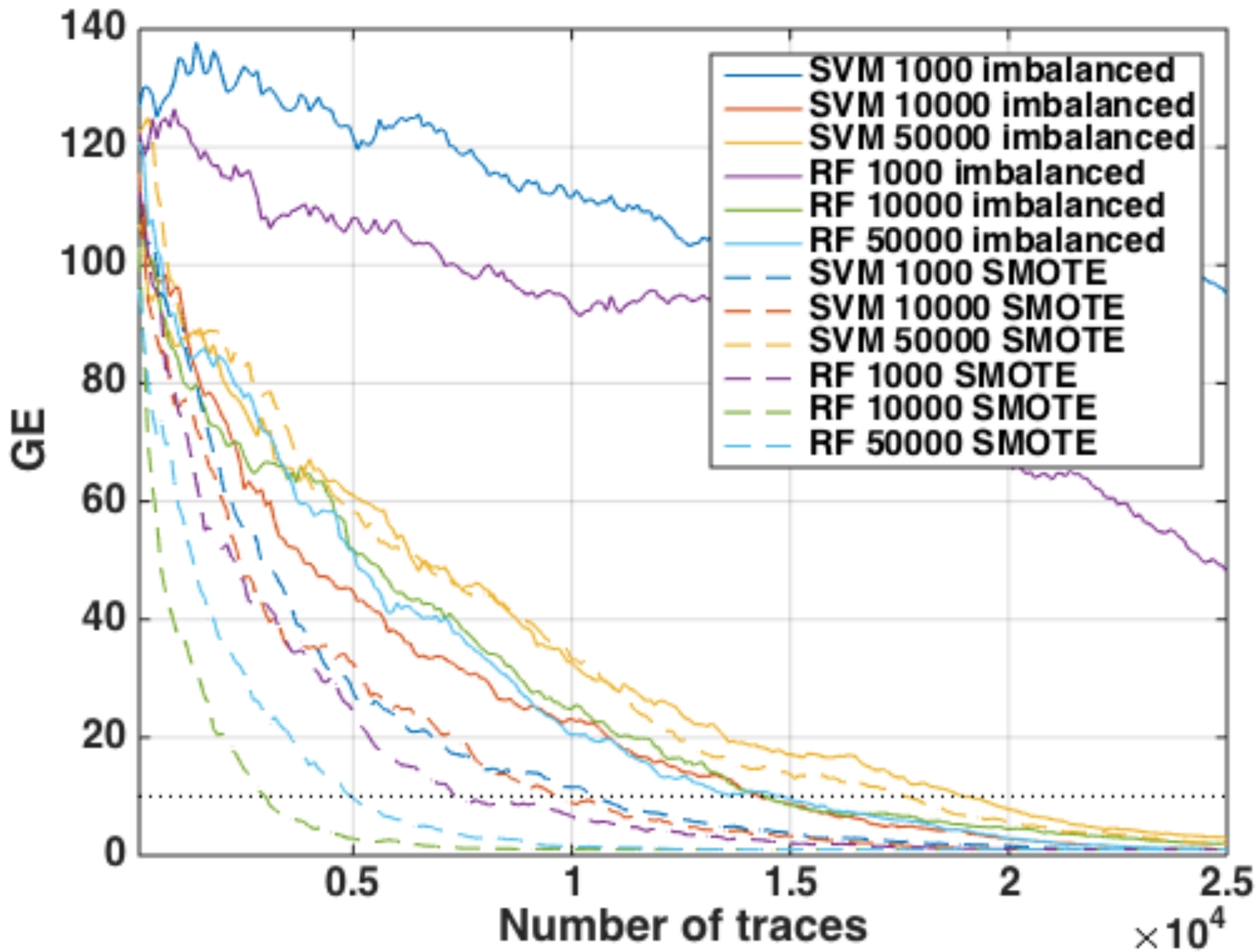
Dataset 2

- high noise dataset
- AES-128 on Xilinx Virtex-5 FPGA of a SASEBO GII evaluation board.
- publicly available on github: https://github.com/AESHD/AES HD Dataset



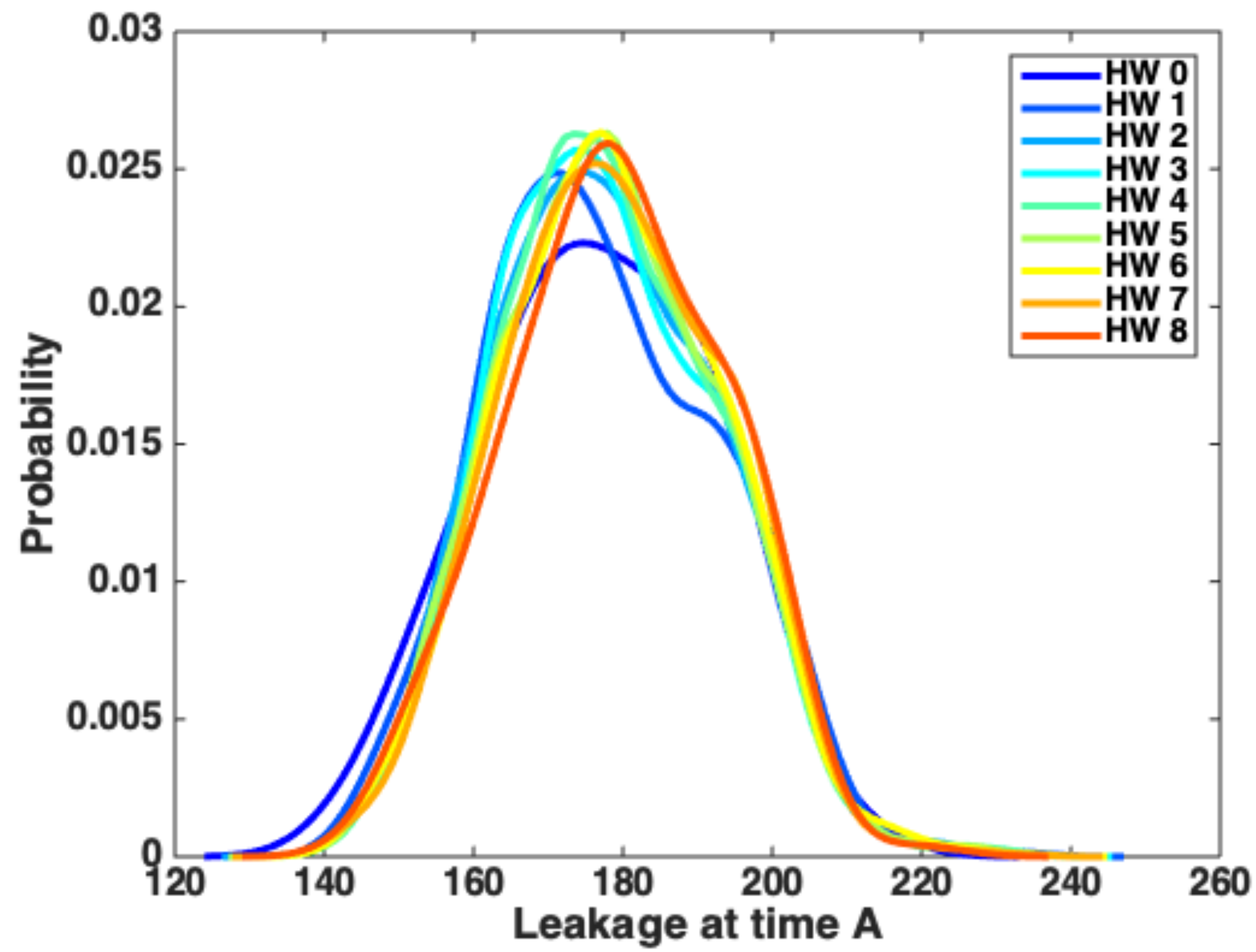
Data sampling techniques

- dataset 2: high noise unprotected



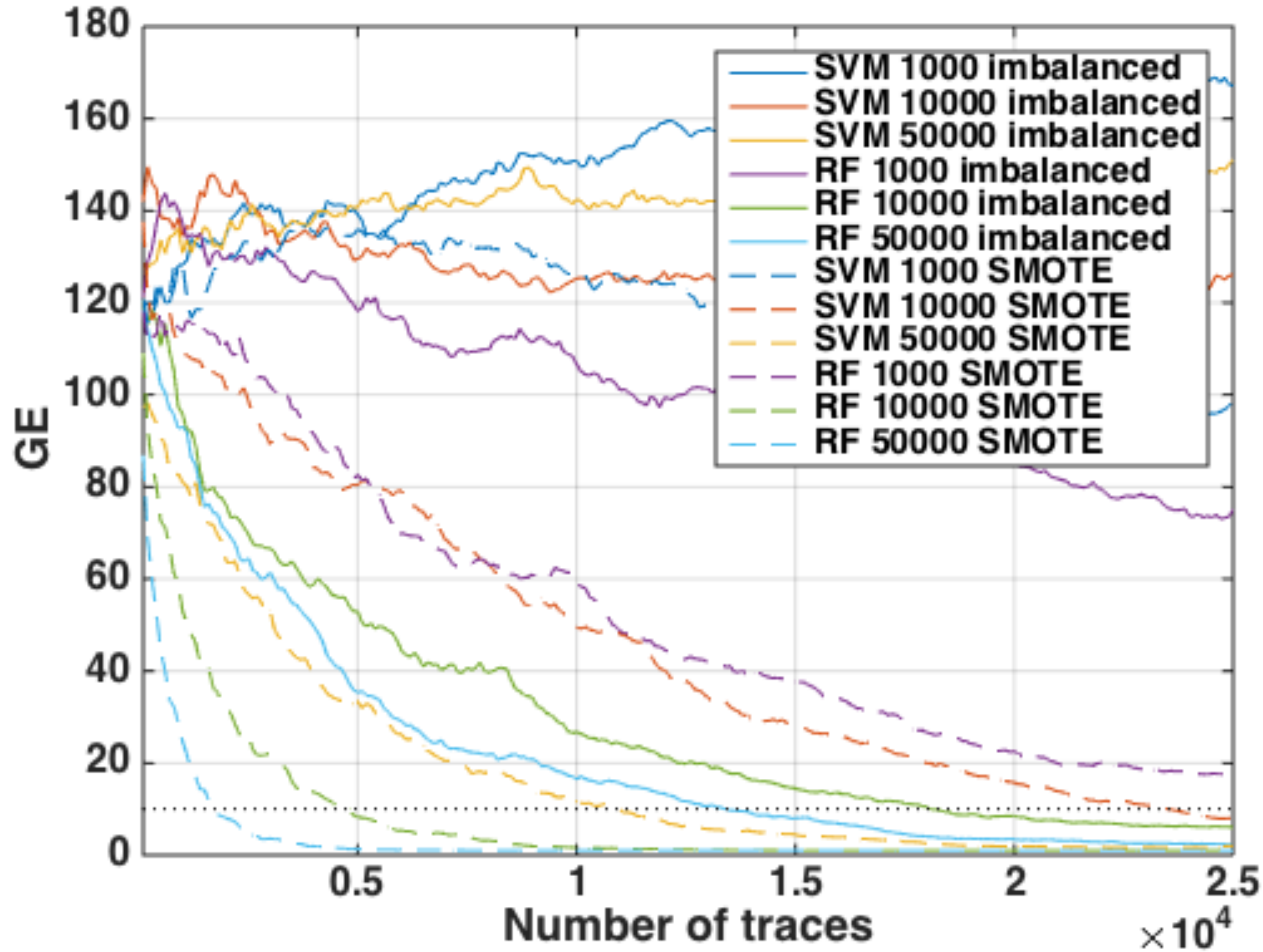
Dataset 3

- AES-128: Random delay countermeasure => misaligned
- 8-bit Atmel AVR microcontroller
- publicly available on github: https://github.com/ikizhvatov/randomdelays-traces



Data sampling techniques

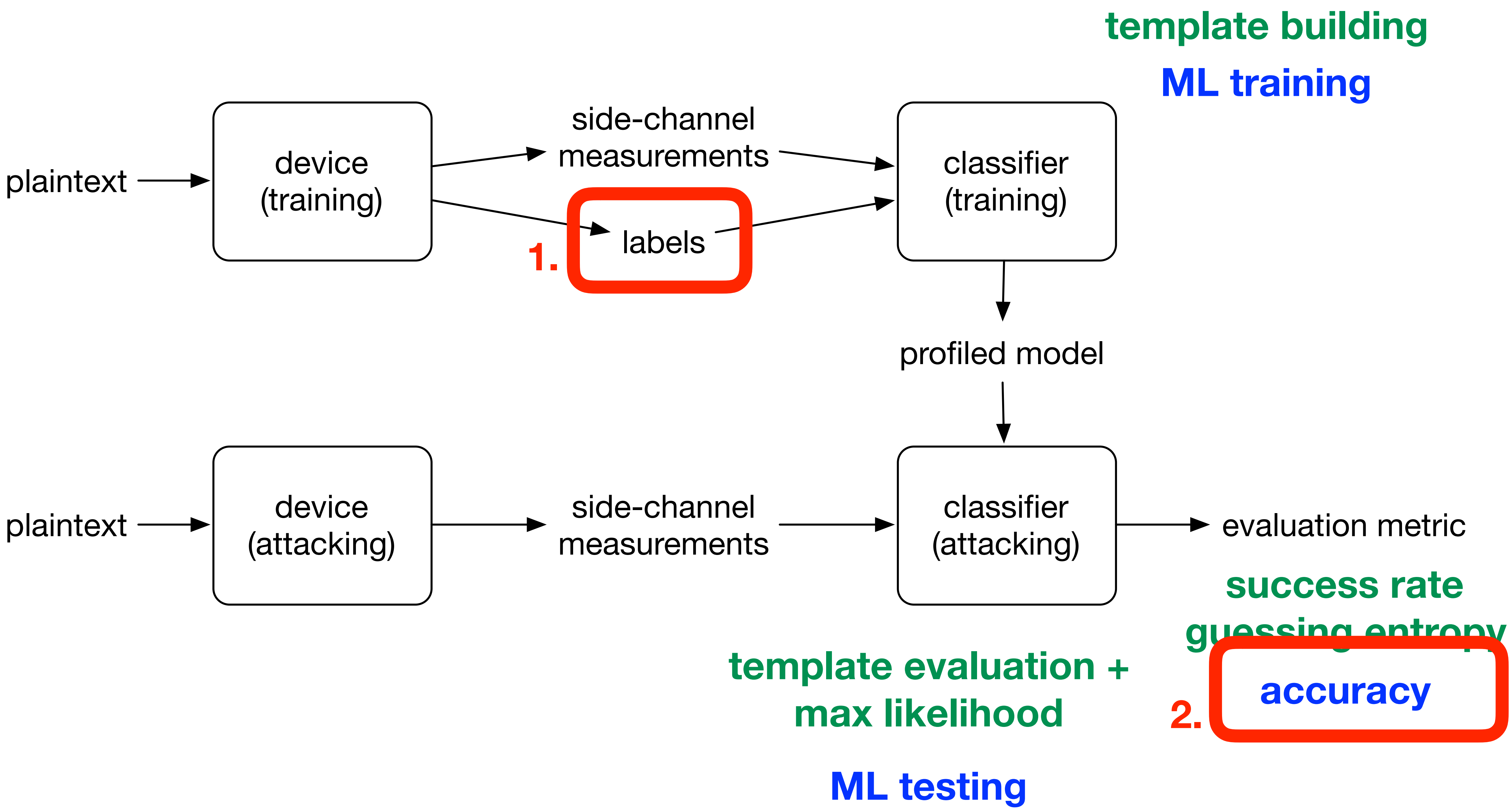
- dataset 3: high noise with random delay



Further results

- additionally we tested SMOTE for CNN, MLP, TA:
  - also beneficial for CNN and MLP
  - not for TA (in this settings):
    - is not “tuned” regarding accuracy
    - may still benefit if #measurements is too low to build stable profiles (lower #measurements for profiling)
- in case available: perfectly “natural”/chosen balanced dataset leads to better performance
- … more details in the paper

Big Picture


template building
ML training
plaintext
device
(training)
side-channel
measurements
1.
labels
classifier
(training)
profiled model
plaintext
device
(attacking)
side-channel
measurements
classifier
(attacking)
evaluation metric
success rate
guessing entropy
template evaluation +
max likelihood
2.
accuracy
ML testing

Evaluation metrics

- SR: average estimated probability of success
- GE: average estimated secret key rank
- depends on the number of traces used in the attacking phase
- **average** is computed over **number of experiments**

- ACC: average estimated probability (percentage) of correct classification
- **average** is computed over **number of experiments**

Evaluation metrics

- SR: average estimated probability of success
- GE: average est… secret key rank
- depends on the number of traces used in the attacking phase
- **average** is computed over **number of experiments**

No translation

- ACC: average estimated probability (percentage) …ect classification
- **average** is computed over **number of experiments**

Evaluation metrics

- SR: average estimated probability of success
- GE: average estimated secret key rank
- depends on the number of traces used in the attacking phase
- **average** is computed over **number of experiments**

- ACC: average estimated probability (percentage) of correct classification
- **average** is computed over **number of experiments**

indication: if acc high, GE/SR should "converge quickly”

SR/GE vs acc

Global acc vs class acc

- relevant for non-bijective function between class and key (e.g. class involved the HW)
- the importance to correctly classify more unlikely values in the class may be more significant than others
- accuracy is averaged over all class values

Label vs fixed key prediction

- relevant if attacking with more than 1 trace
- accuracy: each label is considered independently (along #measurements)
- SR/GE: computed regarding fixed key, accumulated over #measurements
- low accuracy may not indicate low SR/GE

more details, formulas, explanations in the paper…

Take away

- HW (HD) + ML is very likely to go wrong in noisy data!
  - data sampling techniques help to increase performances
  - more effective to collect less real sample + balancing techniques than collect more imbalanced samples
- ML metrics (accuracy) do not give a precise SCA evaluation!
  - ✴ global vs class accuracy
  - ✴ label vs fixed key prediction