WPI

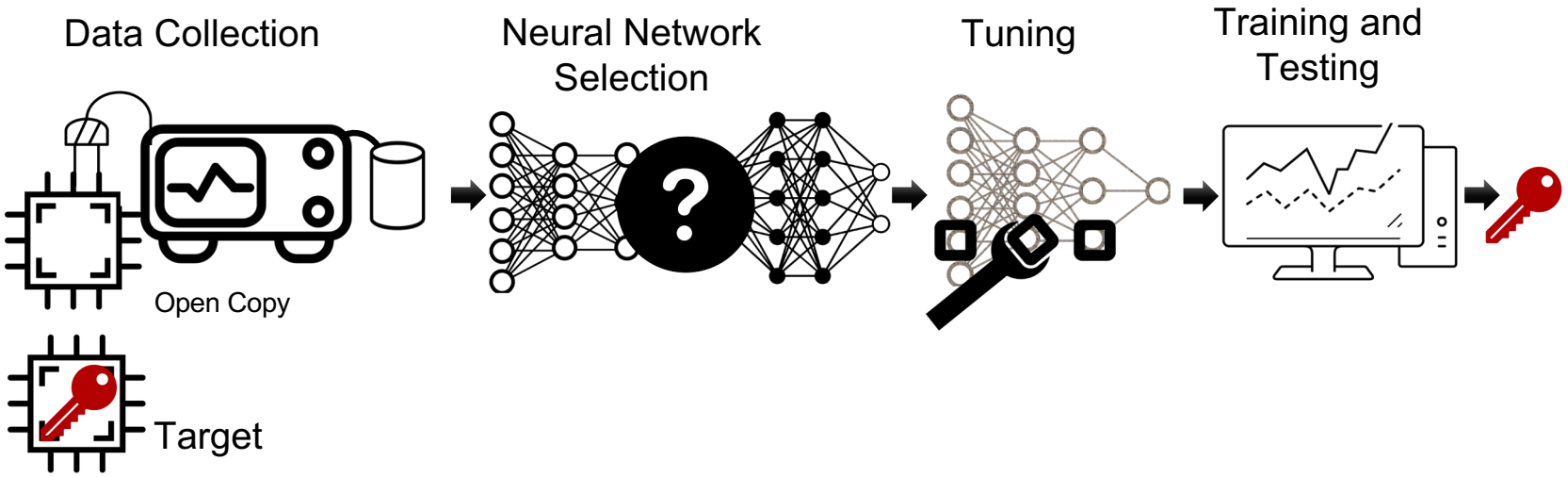
Classic Rewind and Fast-forward to PQC: Lessons Learned from AI-assisted SCA

Fatemeh Ganji, Worcester Polytechnic Institute

Classic Rewind

Profiled SCA

- A target, e.g., a device embodying a cryptographic module such as an advanced encryption standard (AES) crypto-core
- Profiling and attack phases: training and testing sub-tasks in supervised ML
  - Traditionally, characterizing the leakages precisely through statistical techniques
  - NN-assisted profiled SCA: effective against un-/protected cryptographic implementations as well as noisy, and shuffled traces
  - NN-enabled SCA a step toward automating SCA

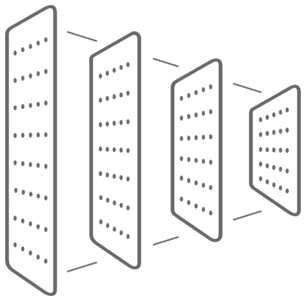


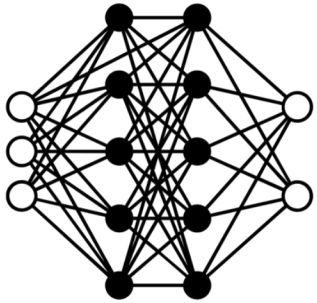
Non-profiled SCA

- Differential Power Analysis (DPA), CorrelationPower Analysis (CPA), or Mutual Information Analysis (MIA)
- Profiled attacks: the most powerful form of SCA
  - Non-Profiled attacks: the attacker is able to collect several traces with a fixed unknown key
- Non-profiled SCA relying on NNs
  - Partition-based side-channel attack [1]
    - DPA-based non-profiled attack
    - For each key guess, the set of traces is partitioned according to guessed intermediate values
    - A statistical distinguisher is used to measure the consistency of each partition and reveal the correct key
    - Role of NNs: evaluate the consistency of the partitions and reveal the correct key by training on the hypothetical intermediate values

[1] Timon, TCHES'19

Which type of NNs?





CNNs vs. MLPs

CNN	MLP
More hyperparameters	More trainable parameters
Feature extraction functionality	Can be equipped with feature selection techniques if needed
More hyperparameters to tune	---
Both doing well when applied in SCA	

Which one to choose?

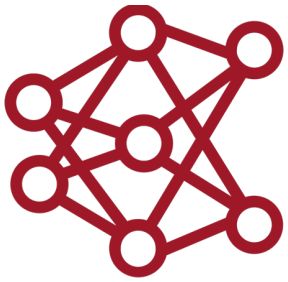
- Occam's razor
  - Choose the simpler
- CNN is good for
  - Text data, time series data, sequence input data
  - A spatial relationship in data
  - One-dimensional: to develop an internal representation of a one-dimensional sequence [1,2]
- MLP
  - Similar data types as for CNNs
  - Do we really need a convolutional layer? [3]
  - Any way to leverage spatial relationships?
    - "Rewiring" fully connected NNs [4]

[1] Zaid, TCHES'19 [2] Wouters, TCHES'20 [3] Wu, IEEE TETC'20 [4] Acharya, TCHES'22

Hyperparameter tuning

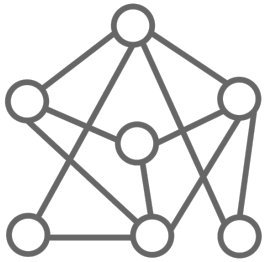
A cautionary note!

Neural Architecture Search (NAS)



- Reinforcement learning
- Visualization techniques
- Layer-level network morphism + Bayesian optimization

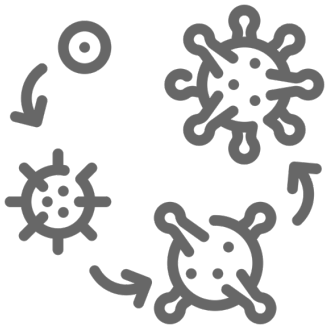
- The (maximum) number of layers: possibly unbounded
- Type of operation for each layer, e.g., pooling and convolution
  - choices for optimizers and loss functions
- Hyperparameters: the number of neurons in a layer of MLPs, and the number of epochs
  - Random search within pre-defined ranges, grid search
  - When to stop training

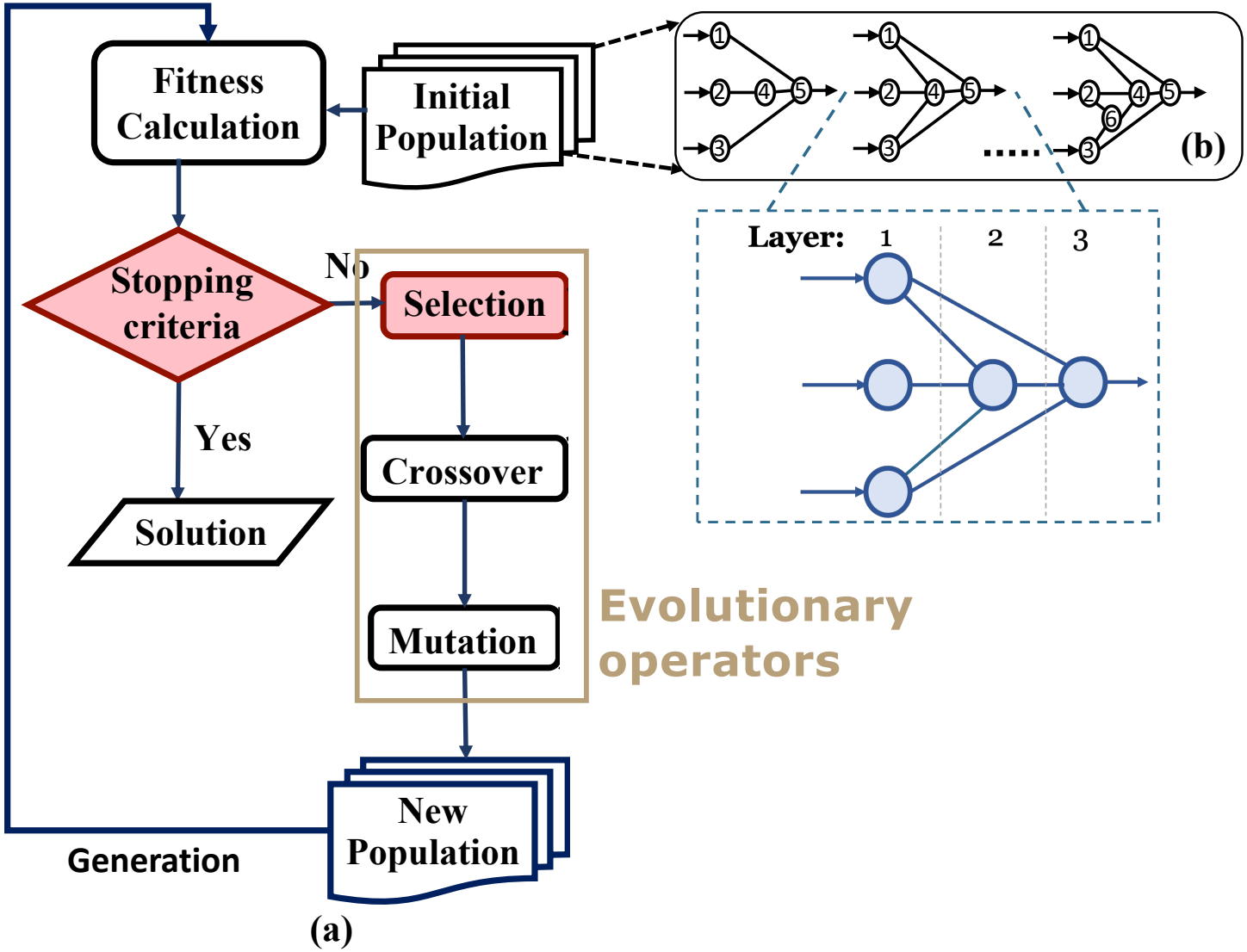


- Multiple branches and skip connections

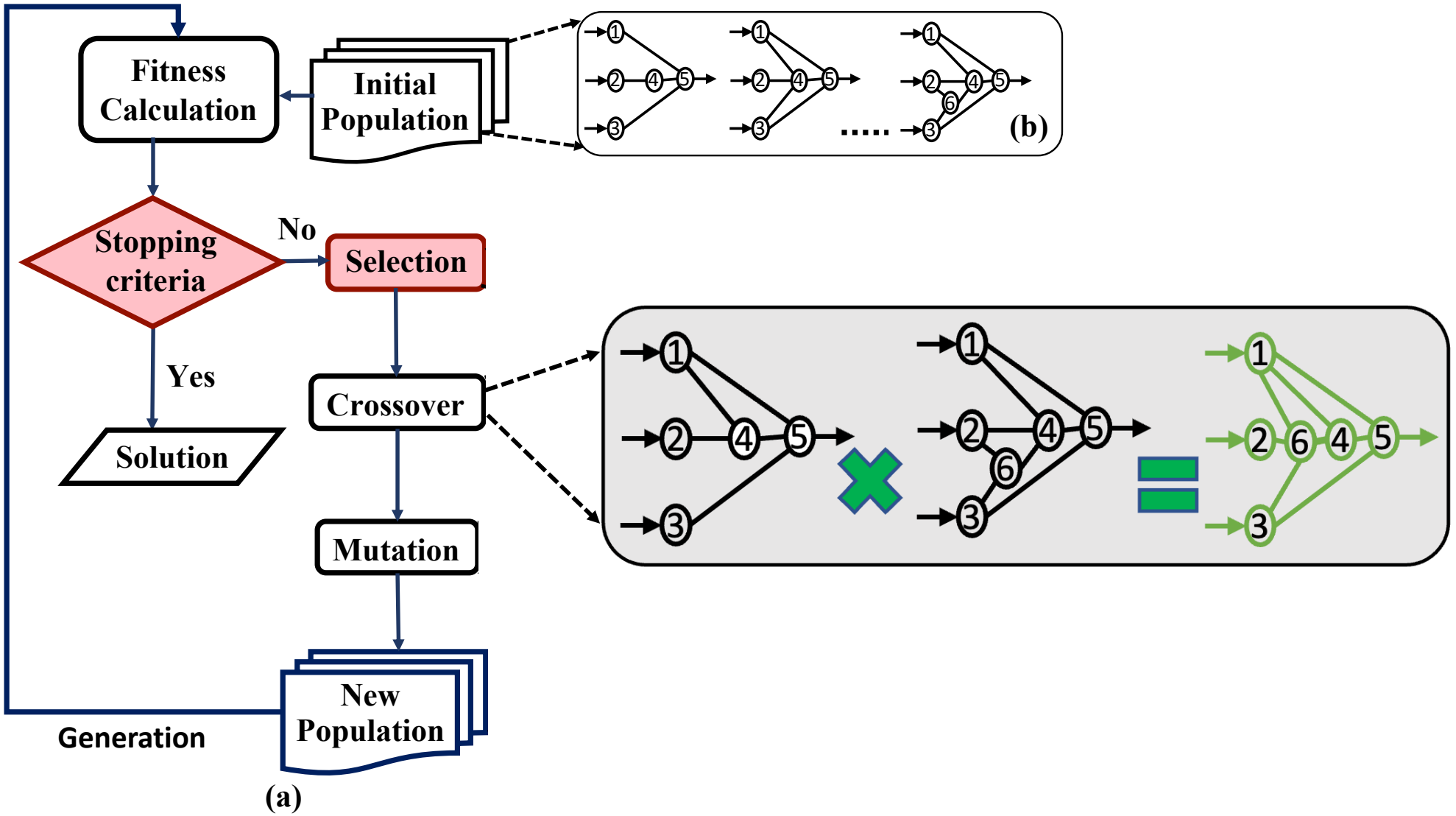
Can we begin with a simple architecture and let it *evolve* to obtain the desired output, e.g., a high accuracy?

Neuroevolution of augmenting topologies (NEAT)

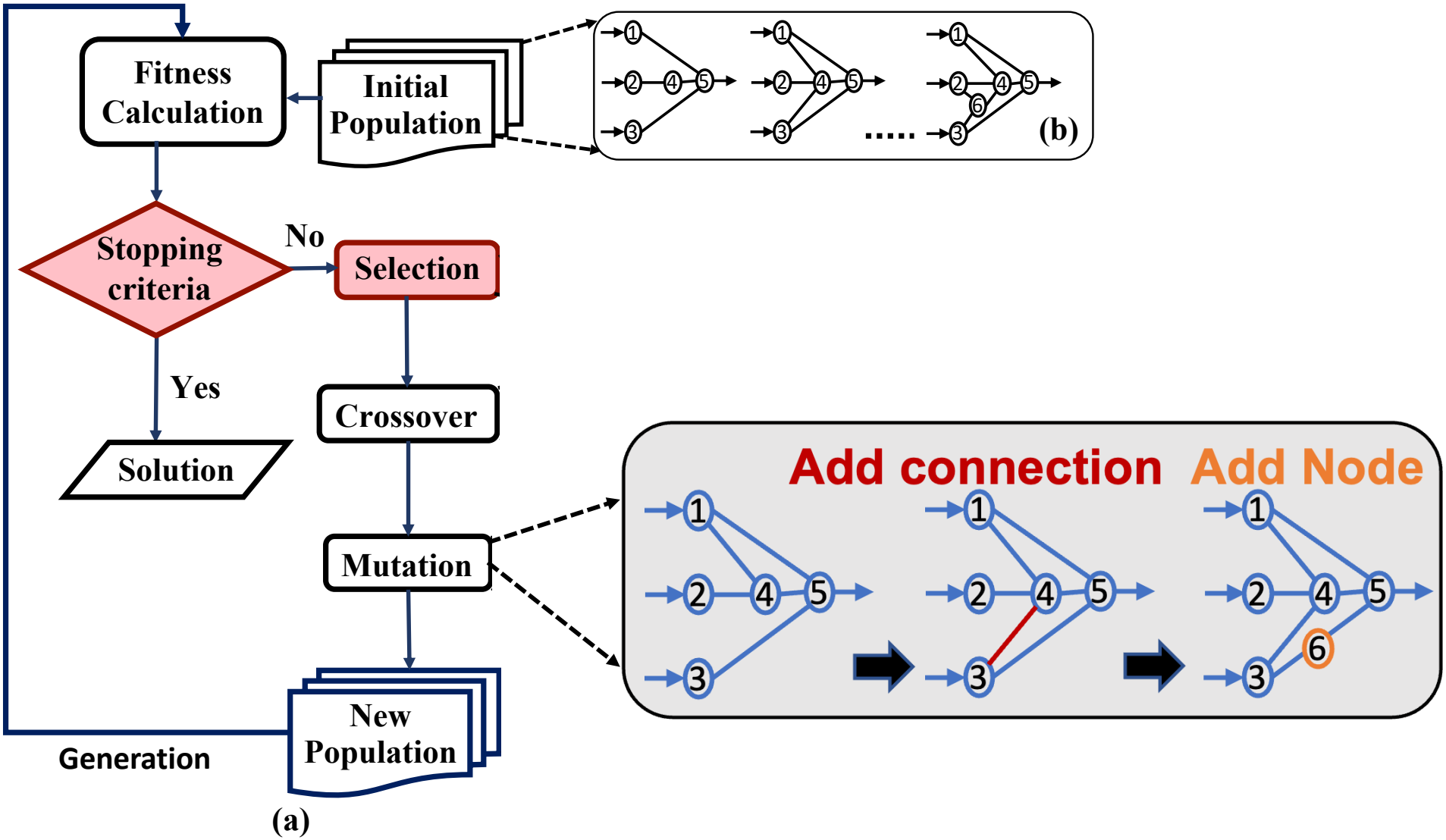




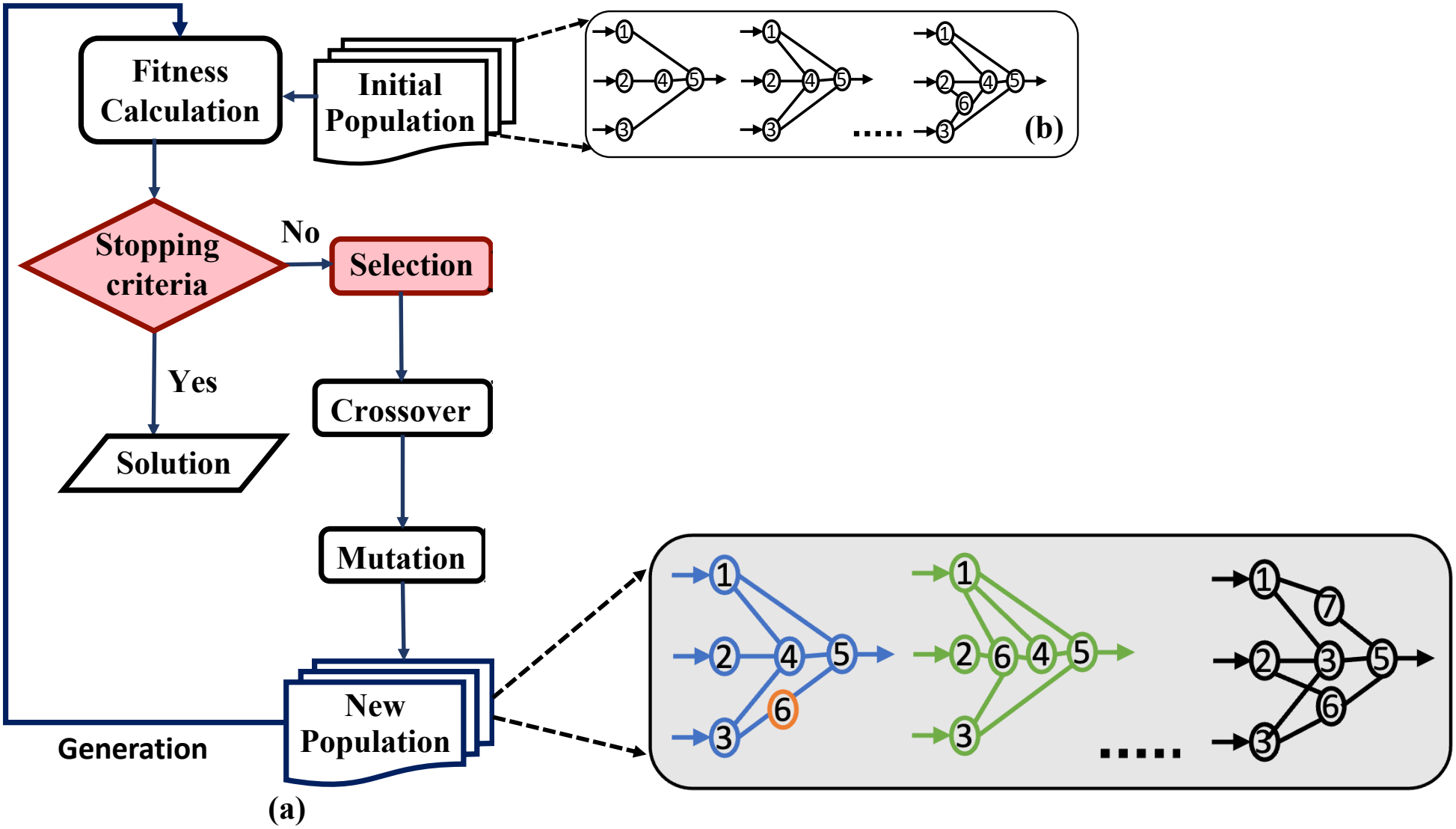
Neuroevolution of augmenting topologies (NEAT)

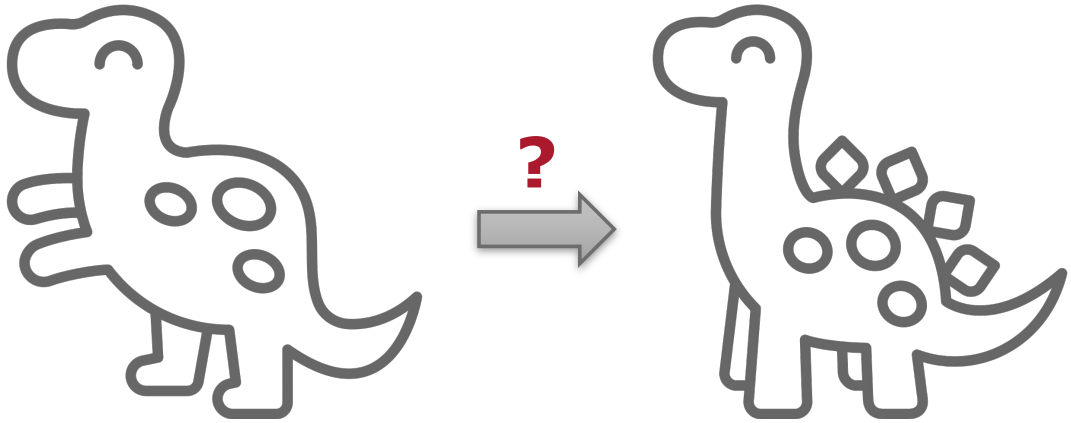


Neuroevolution of augmenting topologies (NEAT)



Neuroevolution of augmenting topologies (NEAT)





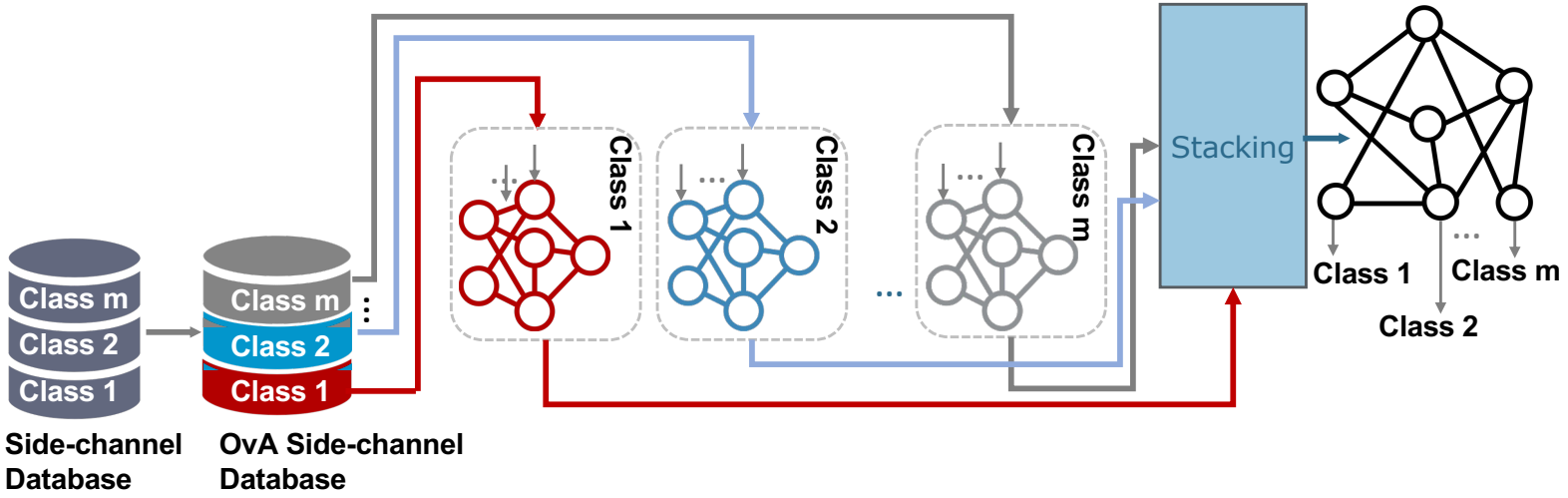
Are all evolutions useful?!

InfoNEAT: Toward making NEAT even better!

- Metric: conditional mutual information (CMI)
  - Intuition: last hidden layer carries the highest level of mutual information between the output of hidden layers and the labels
  - Does adding a new layer/node help?
- Selection criteria
  - Using CMI for last layers of two NNs within a species.
  - If there are multiple networks with similar fitness values, choose the network with lowest CMI value.
  - Helps select appropriate features and deliver an effective well-configured model.
- Stopping criteria
  - Using CMI of two NNs from generation t to t+1
  - If fitness and CMI both do not improve, stop the evolution
  - Reached the lowest possible CMI value

InfoNEAT for SCA

- NEAT cannot handle multi-class tasks
  - One-vs-All (sometimes called "One-vs-Rest") multi-class classification followed by stacking ensemble learning



Results

Experimental setup

- Datasets
  - ASCAD
    - AES-128 protected with 1st-order Boolean masking on an 8-bit AVR microcontroller ATMega8515
    - ASCAD fixed key [1]: 700 features.
    - ASCAD variable key [1]: 1400 features.
    - Additionally, traces desynchronized by 50 and 100 samples window maximum jitter
  - AES_HD [2]: 1250 features.
    - AES-128 on a Xilinx Virtec-5 FPGA
- Setup
  - 8 Skylake Dell C6420CPUs allocated per task and a total memory of 80 GB.
  - No additional model validation step, no plaintext, no hybrid approach
- Metrics
  - Guessing entropy (GE)
  - TGE0: the minimum number of traces to obtain GE=0

[1] https://github.com/ANSSI-FR/ASCAD/tree/master/ATMEGA_AES_v1
[2] https://github.com/AISyLab/AES_HD_2.

Glimpse of the results

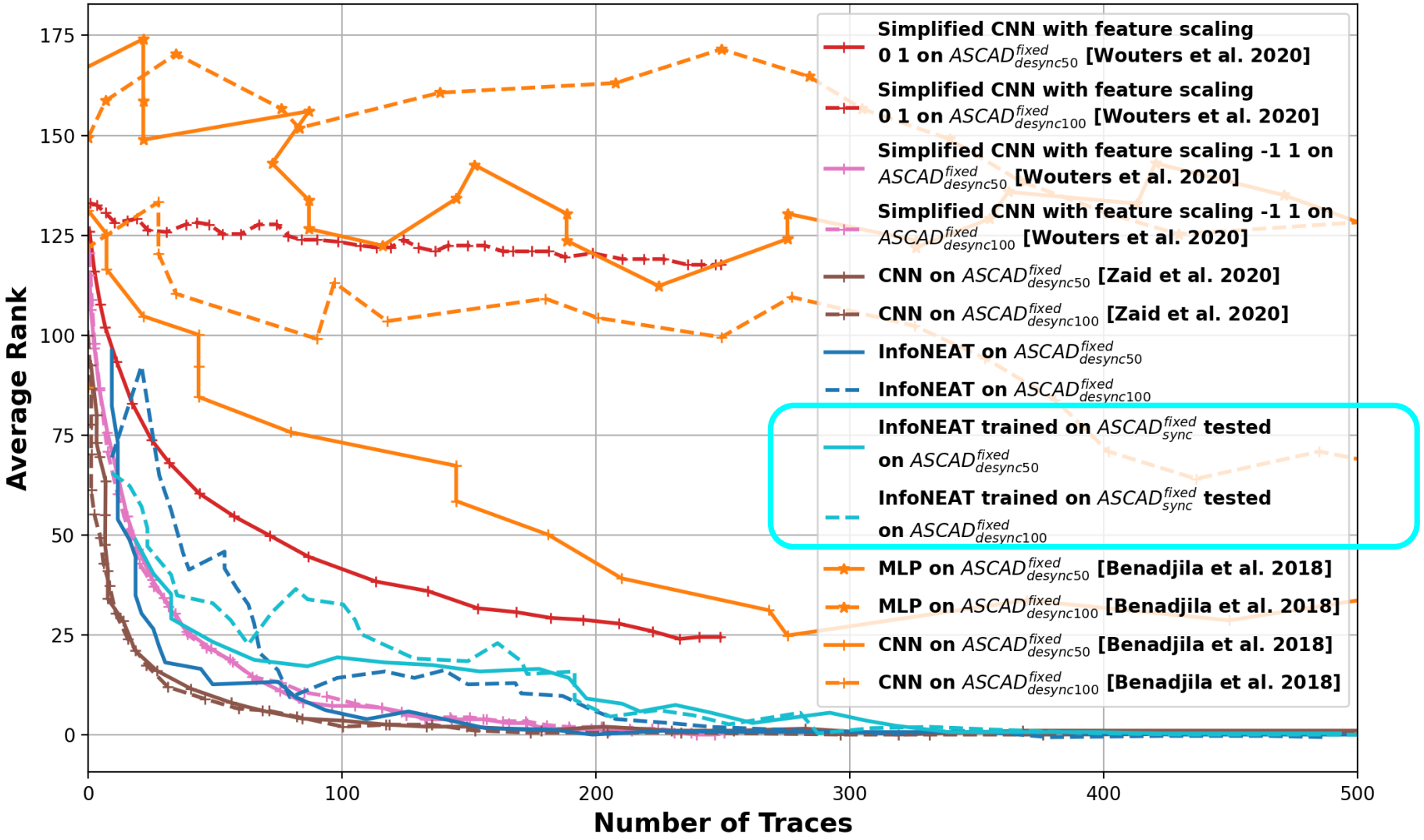
Ref.	# of trainable parameters	Epochs	Training traces	Hyperparameter tuning	Dataset	TGEo
Benadjila et al., 2018	393,936* 66,652,444†	400* 75†	40,000	**Trial and error**	ASCAD fixed	410* 480†
Weissbart et al., 2020	740,136*	200	20,000	**Trial and error**	ASCAD fixed	1630
Wu et al., 2020	478,656*, 54,752†	[10,50]	50,000	**Bayesian optimization, and Random search**	ASCAD fixed	129*/158†
					ASCAD variable	2000* /3144†
Picek et al., 2019	49,024*	-	50,000	**Randomly selected**	AES HD	10,000
Perin et al., 2020	493,480*	-	200,000	**Random Search**	ASCAD variable	56* / 450†
Zaid et al., 2020	16,960†	50	50,000	**Visualization techniques**	ASCAD fixed	191†
	3,282†	20	45,000		AES HD	800†
Wouters et al., 2020	6,436†	50	45,000	**Taken from Zaid et al., 2020**	ASCAD fixed	155†
	2,020†		50,000		AES_HD	800†
Rijsdijk et al., 2021	79,439†	50	50,000	**Reinforcement learning**	ASCAD fixed	202†
	70,492†		200,000		ASCAD variable	490†
InfoNEAT	15,107	8	38,400	**Automatic Neuroevolution**	ASCAD fixed	130
	317,408	8	161,280		ASCAD variable	120
	102,757	33	38,400		AES_HD	170

*MLP, +CNN

Training on synch and testing on desynch datasets

- Not always easy to align traces using alignment techniques: useful to employ ML models robust against desynchronization
- The model is trained against synchronized traces
  - During the profiling phase where the attacker works on an open copy
  - No need to re-train/fine-tune the model, in the presence of desynchronization
- Why is this possible?
  - Feedback connections between neurons automatically inserted using the add connection operator.
  - InfoNEAT adapting to sequences of events, even if they exhibit a lag, e.g., time-delay.

Example: GE for the $ASCAD_{fixed}$ desync dataset



More results…

- Impact of CMI-based criteria
- Training an effective stacked model
- Batch size selection
- Initialization of weights
- Memory and epoch-wise efficiency
  - The number of nodes and species for all the 256 sub-models
- Time-complexity of InfoNEAT training
  - The average number of generations and generation time

Lessons learned

Common Misconception

- More complex networks might lead to better results
- Approaches taken to run NN-enabled SCA more automatically are time-consuming
  - Wall-clock time is relevant!
  - Concrete time complexity through asymptotic notations (big O notation, for instance)
    - Complexity of Grid search [1] > Bayesian optimization [2] > neuroevolution [3]
    - Q-learning algorithm [4] > neuroevolution [3]
- NN-enabled SCA is not generalizable
  - Approaches for characterizing measurements between different devices [5]
- Still in search of loss function?
  - Negative log-likelihood (NLL) and its relationship with perceived information [6]

[1] Perin, TCHES'20 [2] Wu, IEEE TETC'20 [3] Acharya, TCHES'22 [4] Rijsdijk, TCHES'21
[5] Bhasin, NDSS'20 [6] Masure, TCHES'20

Overseen Techniques

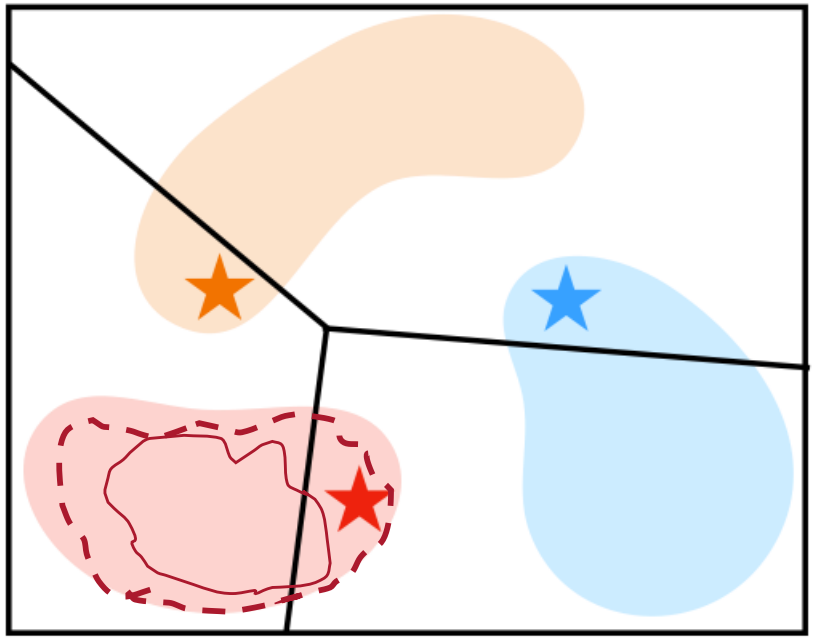
- Bias and variance in NNs
  - High bias: more assumptions about the form of the target function >> persistent error
    - k-fold cross-validation
  - High variance: large changes to the estimate of the target function with changes to the training dataset.
    - Ensemble learning [1,2]
    - Weight initialization
- Overfitting, underfitting, etc.
  - Learning curves
- Validation
  - An extra step to assess the skill of the model
  - The validation dataset MUST be drawn from the training dataset

[1] Perin, TCHES'20 [2] Acharya, TCHES'22

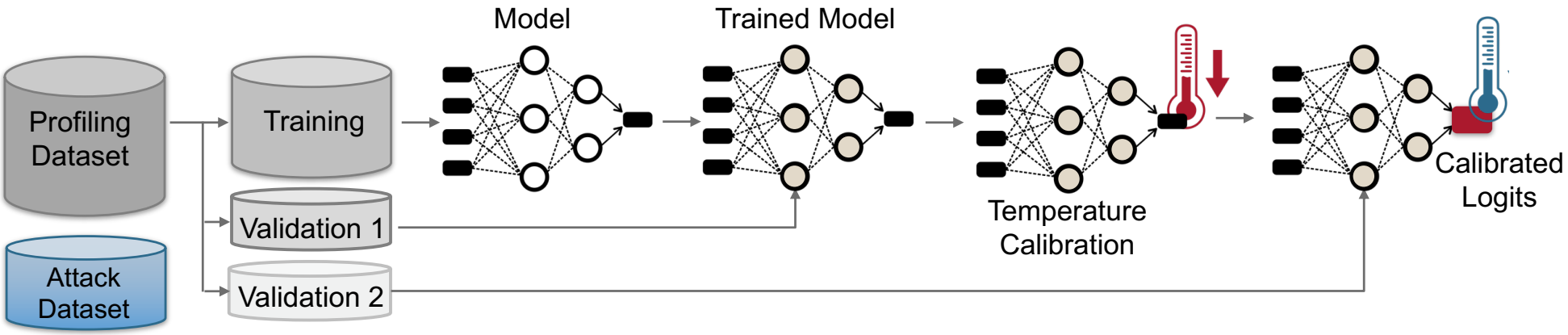
Temperature calibration

How confident are NN-based SCA?

- NNs tend to be "too confident"
  - Confidence: the probability of the correctness of the prediction
  - Calibrate the confidence, i.e., going closer to the true probability
- Temperature Calibration
  - Temperature is a scalar parameter defined in the field of deep learning
  - A metric for judging how well the model is configured and trained before launching the attack



CoolSCA

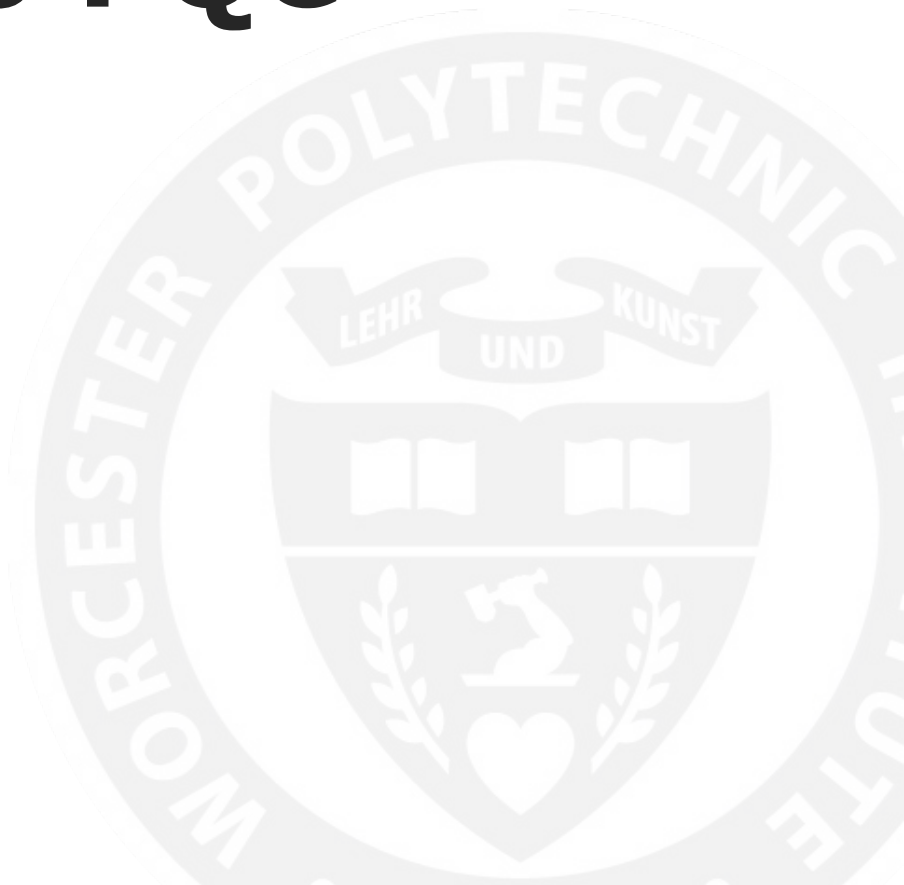


https://github.com/vernamlab/CoolSCA

Some results

Dataset	Model	Leakage model	Uncalibrated		Calibrated
			T	T_{GE0}	T_{GE0}
ASCADf	MLP	ID [WPP22a]	2.953	449	295
		HW [WPP22a]	1.604	432	246
	CNN	ID [WAGP20]	1.513	226	190
		ID [ZBHV20]	1.435	162	120
		HW [PCP20a]	1.605	1922	1895
ASCADr	MLP	ID [WPP22a]	3.08	1492	1137
		HW [WPP22a]	7.216	1152	1164
	CNN	ID [WPP22a]	1.120	347	269
		HW [WPP22a]	1.794	1376	1038
CHES-CTF	MLP	HW [WPP22a]	15.263	2198	768
	CNN	HW [WPP22a]	25.17	5000	4913

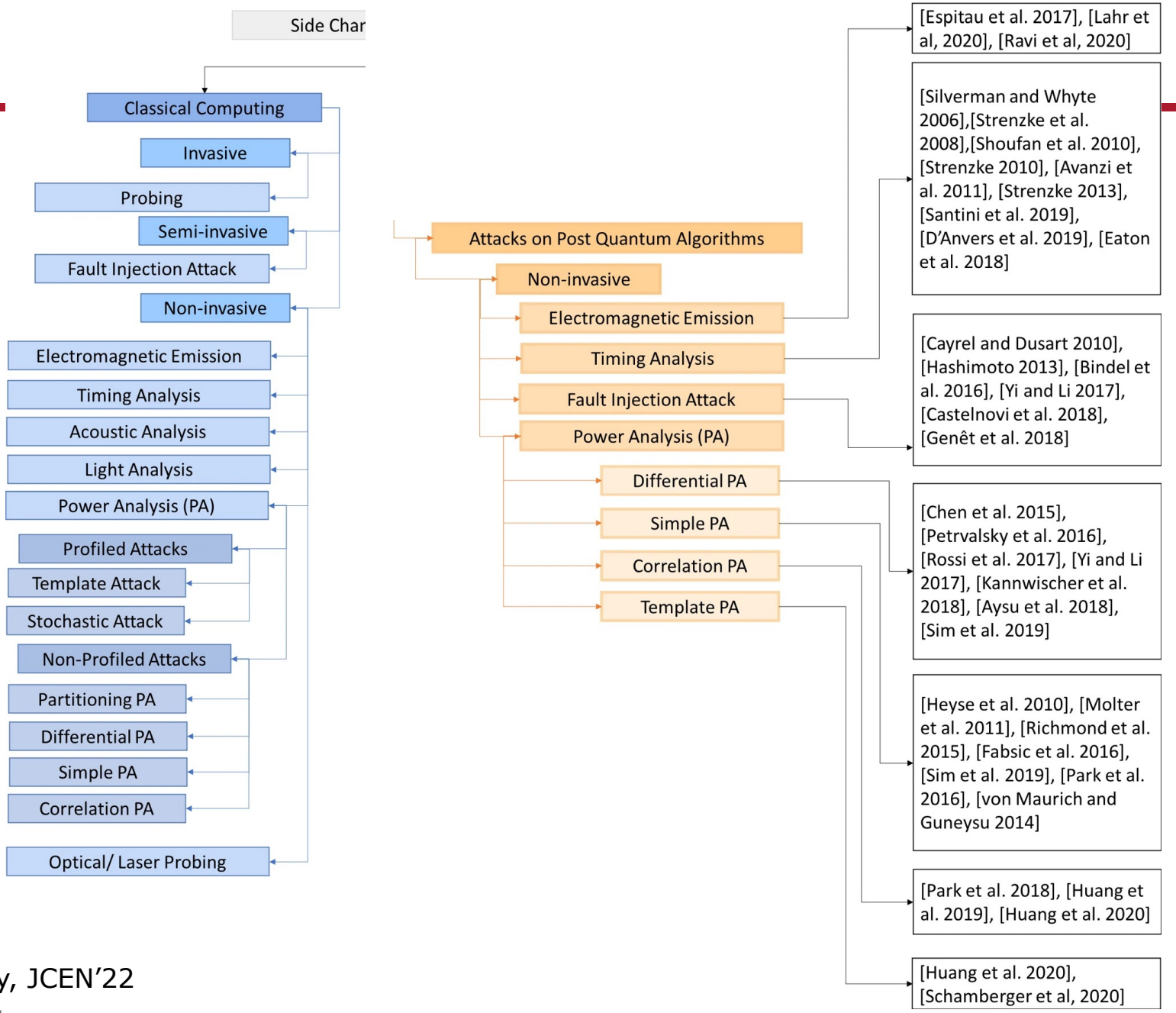
Fast-forward to PQC



SCA against PQC Schemes

- Quantum-resistant cryptographic algorithms
  - Encryption: protecting information exchanged across a public network
    - NIST has selected the CRYSTALS-Kyber
    - Key encapsulation mechanism (KEM)
  - Signatures: identity authentication
    - CRYSTALS-Dilithium, FALCON, and SPHINCS+
    - DPA against SPHINCS

https://sphincs.org/data/sphincs+-r3.1-specification.pdf



[1] Chowdhury, JCEN'22

Examples of attacks against lattice-based PQC

PQC	Target	Method
FrodoKEM	matrix/poly multiplication	CNN
NewHope	matrix/poly multiplication	CNN
Kyber	Message Encoding	MLP
SABER	Message Encoding	MLP
FrodoKEM	Message Encoding	MLP
SABER	POLY2MSG	MLP
SABER	Poly_A2A	Ensembled MLP
SABER	Barrett Reductions	MLP
FrodoKEM	matrix/poly multiplication	CNN
NewHope	matrix/poly multiplication	CNN
SABER	A2B_bitsliced_msg()	MLP
Kyber	FY Indexes	MLP
SABER	FY Indexes	MLP
Kyber	mask_poly_frommsg()	Recursive NN
Kyber	NTT/I-NTT	CPA

[1] Hoang, ISQED'24

NN-enabled SCA against PQC

- Post-quantum KEMs
- Many attacks against Kyber
  - Target procedure (signing/verification), target operation, attack technique, operating mode of Kyber
  - CNNs [1,2], MLP [2]
    - EM traces
    - No information about the choice of the NN, batch size, the number of epoch
    - "…designed to have a sufficient model capacity…"
  - Pushing for higher accuracy

[1] Tanaka, TCHES'23 [2] Ueno, TCHES'21

Challenges

- Lack of available datasets
- Nonhomogeneous devices, APIs, etc.
- Knowledge transfer from attacks in the classic world to post-quantum one
- Finding PoI
  - using the Signal-to-Noise Ratio (SNR) per share -> confusing outcomes [1]

[1] Pay, Africacrypt'24

Conclusion

- InfoNEAT: application of NAS in SCA
  - Tailored toward the needs of SCA
  - Improved NEAT by incorporating information-theoretic metrics
- Temperature as a metric
  - Easy integration into any NN-based SCA
- The gap between approaches explored in ML and SCA
- Where will the NN-enabled SCA against PQC schemes be in the DISTANT future?

Thank You!

InfoNEAT

CoolSCA